



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA INFORMÁTICA EN SISTEMAS DE
LA INFORMACIÓN

Aplicación web para la gestión de apartamentos rurales

Web application for rural apartments management

Realizado por
José María Gómez García

Tutorizado por
Carlos Rossi Jiménez

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2020

Fdo. El/la Secretario/a del Tribunal

Resumen

La siguiente memoria se compone, fundamentalmente, de dos partes. La primera parte, la Introducción, habla sobre los objetivos y la motivación del proyecto, y la solución que se ofrece para el problema a resolver, es decir, recursos software y las tecnologías que han sido utilizadas para la realización y creación de la aplicación desarrollada en este proyecto. La segunda parte es el cuerpo de la memoria, compuesta a su vez de dos partes principales; un análisis de lo que se va a desarrollar, donde cambian un poco los “objetivos” iniciales de la introducción debido a cambios en las funcionalidades en el feedback con el cliente, y por otra parte el desarrollo de la aplicación y una explicación de la estructura del código. Por último, se exponen las conclusiones obtenidas sobre este proyecto y las posibles mejoras que la aplicación podría tener en un futuro.

Palabras clave: Angular, Mongo DB, Reserva de Apartamentos

Abstract

Following document is fundamentally made up by two parts. The first part, the Introduction, talks about the objectives and motivation of the project, and the solution that is offered for the problem to be solved, that is, software resources and the technologies that have been used for the realization and creation of the developed application. The second part is the body of memory, in turn composed of two main parts; an analysis of what is going to be developed, where the initial "objectives" of the introduction change a bit due to changes in the functionalities in the feedback with the client, and on the other hand the development of the application and an explanation of the structure of the source code. Finally, the conclusions obtained on this project and the possible improvements that the application could have in the future are exposed.

Keywords: Angular, Mongo DB, Apartment Bookings

Índice

Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Estructura de la memoria	3
Contexto y descripción general del sistema	5
2.1 Booking	5
2.2 Airbnb	7
2.3 Descripción General del Sistema	11
2.3.1 Módulo de Página Principal - Todos los roles	11
2.3.2 Módulo de Páginas de Apartamentos - Todos los roles	11
2.3.3 Módulo de Reservas de Cliente - Usuarios identificados y administradores	11
2.3.4 Módulo de Panel de Administrador - Usuarios con rol de administrador	12
Requisitos y Casos de Uso	13
3.1 Requisitos Funcionales	13
3.2 Requisitos No Funcionales	15
3.3 Requisitos de Información	15
3.4 Diagrama de Casos de Uso	16
Análisis y Diseño	19
4.1 Maquetación en Balsamiq	19
4.2 Modelado UML	28
4.3 Especificación de Casos de Uso	30
Arquitectura y Entorno Tecnológico	125
5.1 Arquitectura	125
5.2 Entorno Tecnológico	126
5.2.1 Visual Studio Code	126
5.2.2 Mongo DB	129
5.2.3 Angular	131
5.2.4 Node JS	137
5.2.4.1 NPM - Node Package Manager	137
5.2.5 Mongoose + Express	138
Implementación y Pruebas	141
6.1 API	141

6.1.1 Modelos	142
6.1.2 Controladores	143
6.1.3 Servicios	144
6.1.4 Middlewares	145
6.1.5 Rutas	146
6.2 FRONT	148
6.2.1 Modelos	149
6.2.2 Servicios	150
6.2.3 Single Page Application	151
6.2.4 Componentes	153
6.2.5 Vistas	154
6.3 Pruebas	155
6.3.1 Pruebas Unitarias	156
6.3.2 Pruebas de Integración	159
6.3.2.1 Proceso de Comprobación	160
Conclusiones	163
Referencias	165
Manual de Usuario	167
A.1 Usuario no identificado	167
A.1.1 Consultar Oferta	167
A.1.2 Reservar Apartamentos	168
A.1.3 Identificarse como Usuario	169
A.1.4 Visualizar Apartamentos	170
A.1.5 Visualizar Calendario de Disponibilidad	172
A.2 Usuario Identificado	172
A.2.1 Visualizar Reservas Usuario	172
A.2.2 Completar Cuestionario de Satisfacción	173
A.2.3 Realizar Reserva Usuarios Identificados	174
A.3 Usuario Administrador	175
A.3.1 Gestión de Reservas	176
A.3.2 Gestión de Cuestionarios	178
A.3.3 Gestión de Oferta y Anticipos	179
A.3.4 Gestión de Apartamentos	180
A.3.5 Gestión de Tarifas y Periodos	183
A.3.6 Gestión de Disponibilidad	184

1

Introducción

En este capítulo se van a tratar los objetivos y la motivación de este proyecto, así como la solución que se va a dar al problema planteado, es decir, la metodología de trabajo y los recursos software que se utilizarán para la realización del proyecto.

1.1 Motivación

El proyecto surge de la necesidad, por parte de un miembro de mi familia, de gestionar reservas de unos apartamentos situados en el pueblo de Jerez del Marquesado (Granada). Hasta día de hoy, la gestión se ha llevado a cabo mediante hojas de excel para tener todos los datos de las reservas y clientes, un trabajo tedioso ya que no existe una base de datos centralizada.

Por otra parte, de forma personal, el hecho de realizar una aplicación web completa con tecnologías punteras es algo que me llama bastante la atención, ya que la parte de programación web es en la que quiero especializarme tras acabar la carrera.

1.2 Objetivos

Este proyecto consistirá en una aplicación web que permita una gestión completa de apartamentos rurales. La aplicación contendrá una base de datos y un diseño basado en el patrón modelo-vista-controlador.

Las funcionalidades que se desarrollarán como objetivo principal son las siguientes:

- Página principal de presentación

- CRUD de apartamentos: El usuario podrá añadir nuevos apartamentos añadiendo fotos, habitaciones, texto, etc.

- Procedimiento de Petición de Reserva: Los clientes podrán realizar una petición de reserva de apartamentos introduciendo una fecha de entrada y otra de salida, de forma que el sistema les diga qué apartamentos están disponibles.

- CRUD de Tarifas: El usuario podrá crear nuevas tarifas que se aplicarán cuando un cliente trate de hacer una petición de reserva.

- Calendario de Tarifas y Disponibilidad: Habrá un calendario por apartamento y tarifa, de manera que cada apartamento tendrá su calendario de disponibilidad por fecha y tarifa aplicada. El usuario podrá desactivar días concretos de forma que no estén disponibles para reserva así como cambiar el precio de una noche y tarifa concretas.

- Gestión de Reservas: El usuario deberá de confirmar las peticiones de reserva que han hecho los clientes, valorar una reserva una vez transcurrida la estancia en función del estado del apartamento, buscar reservas introduciendo el DNI de un cliente y gestionar las formas de pago disponibles en función de la fecha introducida.

- Cuestionario satisfacción de Reservas: Los huéspedes, una vez terminada la reserva, tendrán la opción de realizar un cuestionario de satisfacción. Nuestro cliente podrá visualizar dichos cuestionarios mediante su panel de control.

- CRUD Ofertas: El usuario podrá crear ofertas personalizadas en fechas concretas para atraer a los clientes, que podrán visualizarlas en un apartado de la web.

- CRUD Clientes: El usuario podrá crear cuentas para clientes a través del panel de control, de forma que estos puedan iniciar sesión en la página web, acceder a sus reservas, completar cuestionarios de satisfacción e incluso la posibilidad de disfrutar de descuentos dependiendo de las valoraciones de sus reservas anteriores por parte del usuario.

Además de los citados objetivos específicos, el TFG tendrá como objetivos generales la familiarización con el desarrollo de aplicaciones web, poner en práctica técnicas adquiridas durante la carrera y ampliar estos conocimientos aprendiendo nuevas desarrollo como Angular, un potente framework muy usado hoy día y requerido en puestos de trabajo.

Para la elaboración del proyecto se utilizarán las siguientes tecnologías:

- Angular para la parte de front-end
- Java con Spring Boot para la parte back-end
- MongoDB como sistema gestor de base de datos
- Eclipse para el entorno de trabajo
- Visual Studio Code para desarrollar el código fuente de la parte de front-end

1.3 Estructura de la memoria

La estructura de la memoria se divide de la siguiente manera:

- **Contexto y descripción general del sistema:** En este apartado se estudiarán otras aplicaciones parecidas en el mercado, haciendo una breve comparación final con respecto a la que vamos a desarrollar y finalizando con una descripción general del sistema que se va a implementar.
- **Requisitos y Casos de Uso:** Analizaremos requisitos funcionales del sistema, así como requisitos no funcionales y requisitos de información, que posteriormente nos servirán para continuar con el análisis y diseño de la aplicación. Además, incluiremos un diagrama de casos de uso basado en los requisitos funcionales.
- **Análisis y Diseño:** Realizaremos una maquetación de la aplicación, procederemos con el modelado UML con el diagrama de clases y el modelo físico de datos y finalmente especificaremos los casos de uso del apartado de requisitos, añadiendo un diagrama de secuencia para cada uno de ellos.
- **Arquitectura y entorno tecnológico:** Analizaremos las herramientas tecnológicas utilizadas para este proyecto, destacando algunas de sus utilidades más importantes.
- **Implementación y pruebas:** Procederemos a escribir implementación de la aplicación y explicaremos la estructura del código de la aplicación. Esto nos permitirá entender mucho mejor el código y el sistema de carpetas.
- **Conclusiones:** Realizaremos una conclusión final sobre el trabajo realizado, valorando puntos importantes y algunas dificultades que se han encontrado durante el desarrollo del proyecto.

2

Contexto y descripción general del sistema

En este capítulo se va a realizar un estudio del contexto en el que situamos la temática de la aplicación analizando otras aplicaciones similares, tanto en aspecto funcional como no funcional.

A día de hoy existen muchas webs donde los usuarios pueden promocionar una vivienda para alquilarla, pero sin duda existen dos por excelencia; Booking & Airbnb. A continuación evaluaremos dichas aplicaciones y analizaremos sus características más importantes desde los puntos de vista funcional y no funcional.

2.1 Booking

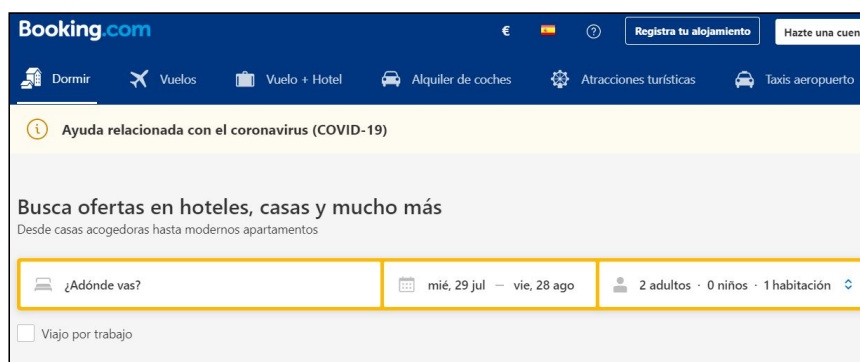


Figura 1: Vista principal de Booking

Es sin duda la web de alquiler más relevante, más conocida internacionalmente y de donde llegan más reservas por parte de los clientes. Tenemos dos roles para usar la web: como propietario y como cliente.

Como cliente, el sistema te deja buscar un alojamiento al mismo tiempo que vuelos, coches o taxis en el mismo aeropuerto de llegada, entre otros (Figura 1).

¡Da de alta tu alojamiento en Booking.com y empieza a recibir clientes cuanto antes!

Para empezar, elige el tipo de alojamiento que quieres dar de alta en Booking.com

Inicio rápido





Apartamento	Casas	Hotel, B&B y similares	Otros
 Apartamento Alojamiento amueblado e independiente que se puede alquilar entero.	 Casas Alojamientos como casas, apartamentos, villas, etc.	 Hotel, B&B y similares Alojamientos como hoteles, bed and breakfasts, hostales, pensiones, albergues, apartahoteles, etc.	 Otros Alojamientos como barcos, campings, campings de lujo, etc.
Da de alta tu alojamiento	Da de alta tu alojamiento	Da de alta tu alojamiento	Da de alta tu alojamiento

Figura 2: Primer paso para dar de alta tu alojamiento en Booking

Por otra parte, como propietario puedes alojar tu propia vivienda para alquilarla (Figura 2). Una vez rellenas los datos básicos sobre tu alojamiento como localización, imágenes y habitaciones, el sistema te pide un precio por noche, y vemos que la plataforma se queda con un 16,10% de la reserva en comisiones y cargos (Figura 3).

¿Cuánto quieres cobrar por noche?

Precio que pagan los clientes

€ 23,00

Incluye impuestos, cargos y comisión

☐ Rebaja tu precio un 20%

Consigue tus primeras reservas más rápido rebajando un poco el precio al principio. Puedes volver a subirlo cuando quieras.

16,10% Cargos y comisión de Booking.com

- ✓ Asistencia en tu idioma las 24 horas
- ✓ Ahorra tiempo con las reservas confirmadas automáticamente
- ✓ Anunciamos tu alojamiento en Google
- ✓ Nos encargamos de los pagos de los clientes ⓘ

€ 19,30 Lo que ganas (incluye impuestos)

Figura 3: Configuración de precio de alojamiento como propietario en Booking

También tiene un sistema de cancelación configurable por el cliente (Figura 4).

¿Cuántos días antes de la llegada pueden **cancelar gratis la reserva** los clientes?

Recomendado

1 día	5 días	14 días	30 días
-------	--------	---------	---------

Protección frente a reservas hechas por error

☐ Off

Para que no pierdas tiempo gestionando reservas hechas por error, eliminamos los cargos de cancelación automáticamente si el cliente cancela en un plazo de 24 horas después de hacer la reserva.

Figura 4: Configuración del sistema de cancelación como propietario en Booking

2.2 Airbnb

Alojamientos Estancias largas Experiencias Experiencias online

UBICACIÓN ¿A dónde viajas?	LLEGADA Introduce las fech...	SALIDA Introduce las fech...	HUÉSPEDES Añade los huéspe...	
-------------------------------	----------------------------------	---------------------------------	----------------------------------	--

Figura 5: Vista principal de Airbnb

Junto con Booking, Airbnb es la web para encontrar alquiler de alojamientos más usada por los clientes en España. De la misma forma, Airbnb puede ser usado como rol de cliente o como rol de propietario.

Como cliente, el sistema no solo te permite buscar alojamiento, sino que también te da la opción de buscar directamente experiencias, tales como una visita a un museo u otra actividad a realizar en la ciudad y combinarlo con tu alojamiento (Figura 6).



Figura 6: Buscador de experiencias de Airbnb

Como propietario puedes tanto organizar tu propia experiencia como publicar, no solo alojamientos enteros, sino también una habitación privada o una habitación compartida (Figura 7), gran característica que la diferencia de Booking.

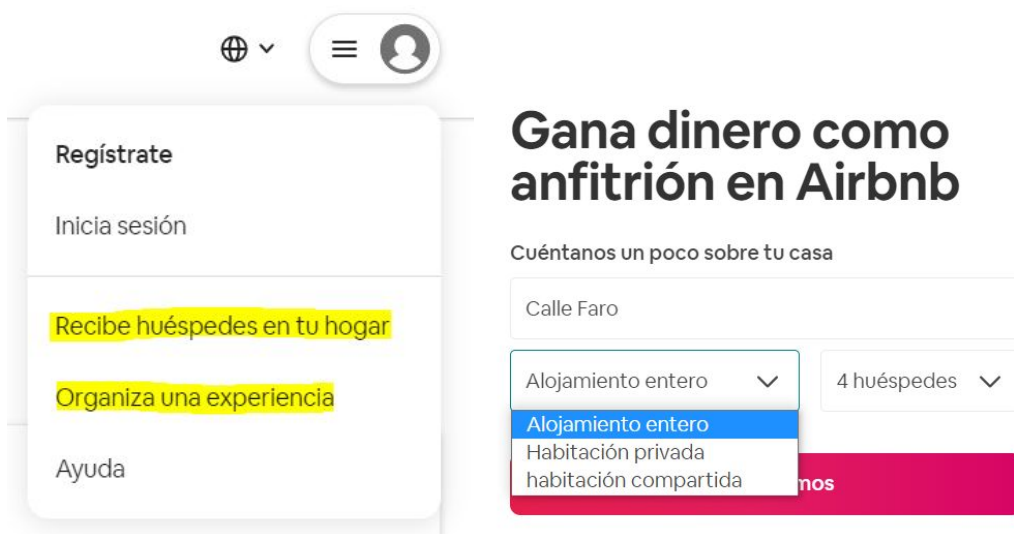


Figura 7: Opciones de publicación como anfitrión en Airbnb

Bajo el alojamiento entero, que es lo que estudiamos para este trabajo, se encuentran otras subcategorías como vivienda anexa, casa, apartamento, hotel boutique, entre otros (Figura 8).

Apartamentos

Los apartamentos son viviendas habitualmente ubicadas en edificios o complejos residenciales de varios bloques.

Casa

A menudo son estructuras independientes, pero algunas, como los dúplex, pueden ser adosadas o compartir las zonas exteriores

Vivienda anexa

Tienen una entrada independiente para los huéspedes, pero están situadas dentro de la misma propiedad de otra residencia principal

Alojamiento único

Alojamientos con características estructurales interesantes o poco convencionales en comparación con las casas y los apartamentos tradicionales

Bed and breakfasts

Empresas de la industria hotelera que ofrecen desayuno a sus huéspedes. Generalmente, los anfitriones viven en la propiedad.

Hotel boutique

Empresas de la industria hotelera con un estilo único y decoración propia, a menudo temática, que los define y diferencia unos de otros.

Figura 8: Subcategorías de alojamiento entero en Airbnb

La web te permite multitud de opciones a configurar tales como reglas de la casa, hora de llegada, si la vivienda incluye pertenencias, si se permite fumar, el tiempo que necesitas antes de que lleguen los clientes para poder limpiarla (Figura 9), la antelación máxima con la que puede hacerse una reserva, noches mínimas y máximas, etc.

¿Cuánto tiempo necesitas para tener todo listo antes de que lleguen los huéspedes?

☐

 El mismo día

☐

 1 día

☒

 2 días

☐

 3 días

☐

 7 días

Hoy

5

6

7




Figura 9: Opciones de margen entre alquileres en Airbnb

Además, permite al usuario bloquear días manualmente o meses enteros, en caso de que la vivienda no esté disponible para alquilar (Figura 10).

enero 2021				Marca este mes como disponible		
lun.	mar.	mié.	jue.	vie.	sáb.	dom.
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Figura 10: Opciones de margen entre alquileres en Airbnb

A la hora de poner precio a la vivienda, el sistema te da la opción de hacer un 20% de descuento a los primeros huéspedes para dar a conocer tu alojamiento, así como ofrecer descuentos por reservar 1 semana o 1 mes (Figura 11).

Algo especial para tus primeros huéspedes

Ofrece un descuento del 20% a tus primeros huéspedes

RECOMENDADO


Los primeros 3 huéspedes que reserven tu alojamiento obtendrán un 20 % de descuento en el precio por noche. Esta oferta especial te ayudará a atraer a más huéspedes y a conseguir las 3 reseñas necesarias para obtener una valoración con estrellas.

No quiero agregar una oferta especial

Una vez que publiques tu anuncio, no podrás agregar esta oferta.

< Atrás

Siguiente



¿Por qué debería agregar una oferta especial?

Con esta oferta podrás atraer a tus primeros huéspedes y recibir tus primeras reseñas. Una vez que consigas tres reseñas, recibirás una valoración con estrellas en los resultados de búsqueda y esto ayudará a que tu anuncio se destaque. Los nuevos anuncios que consiguen reservaciones y reseñas durante el primer mes suelen aparecer con más frecuencia en los resultados de búsqueda y los reservan en promedio 3,6 veces más durante los tres primeros meses. ?

<

● ● ●

>

Figura 11: Opciones de ofertas especiales en Airbnb

Por último, comentar que Airbnb cobra un 3% de tarifa de servicio para anfitriones por cada reserva recibida, mucho más barato que mediante booking.

2.3 Descripción General del Sistema

Una vez realizado un estudio y comparación de las webs de alojamientos turísticos más relevantes procedemos a describir brevemente el proyecto a realizar. Hay que aclarar que la web, a diferencia de las que hemos estudiado, no pretende competir con ellas, puesto que es un proyecto para un cliente personal y, por lo tanto, solo habrá un único anfitrión de apartamentos.

La web se compondrá de cuatro módulos principales; página principal y de reservas, páginas de apartamentos, página de reservas de clientes y panel de administrador. Por otra parte, la página podrá usarse mediante tres roles de usuario; usuario no identificado, usuario cliente y usuario administrador. Los dos primeros módulos corresponden con el rol de usuario no identificado, el módulo reservas de clientes es accesible para usuarios clientes y, por último, el módulo de panel de administrador es sólo accesible para usuarios administradores.

2.3.1 Módulo de Página Principal - Todos los roles

Por un lado, en la página principal tendrá el procedimiento de reserva, en el que los usuarios introducirán las fechas y el sistema le ofrecerá las posibilidades, precio y demás, y se completará la reserva creando un usuario nuevo en caso de que no esté registrado. Además, tendrá una vista del calendario y su disponibilidad y un pie de enlace a las páginas de los apartamentos.

2.3.2 Módulo de Páginas de Apartamentos - Todos los roles

En el módulo de las páginas de apartamentos aparecerán los apartamentos y sus enlaces a las páginas individuales de cada una, donde se podrán ver descripción, título, fotos en un slider, camas y habitaciones.

2.3.3 Módulo de Reservas de Cliente - Usuarios identificados y administradores

En el módulo de reserva de clientes, sólo disponible si el usuario está conectado y administradores, los clientes podrán ver las reservas que han realizado, el estado en el que se encuentran y, en caso de haber pasado la fecha de la reserva, la opción de completar un cuestionario de satisfacción.

2.3.4 Módulo de Panel de Administrador - Usuarios con rol de administrador

En el módulo de administrador, solo disponible para los usuarios con el rol administrador, se podrá acceder a todas las reservas de la base de datos y su información, se podrán visualizar los cuestionarios de satisfacción rellenados por los clientes, añadir ofertas y fechas con anticipo, modificar los datos de los apartamentos o añadir nuevos apartamentos, añadir nuevas tarifas y periodos para los apartamentos y comprobar la disponibilidad del calendario, con la diferencia con respecto a la página principal de que además se podrán añadir fechas en las que ciertos apartamentos se encuentren ocupados manualmente.

Requisitos y Casos de Uso

En este capítulo vamos a analizar los requisitos necesarios para implementar la aplicación, analizando qué es lo que quiere el cliente exáctamente. Además, diseñaremos un diagrama de casos de uso que definirá qué puede hacer cada usuario dentro de la aplicación en función del rol.

Tras habernos sentado con el usuario y tener una larga conversación acerca de lo que quiere exactamente, concluimos 31 requisitos en total, que cubren todas las necesidades. Se dividen en 16 requisitos funcionales, 6 requisitos no funcionales y 9 requisitos de información.

3.1 Requisitos Funcionales

AR-RF-01 - Reservar apartamentos: En la página principal debe haber un procedimiento de petición de reserva en el que los futuros huéspedes puedan introducir la fecha en la que quieren hospedarse y el sistema le ofrezca las posibilidades a escoger. Cuando seleccionen los apartamentos que quieren reservar, el sistema comprobará si la fecha requiere anticipo o no y les mostrará el número de cuenta al que deben ingresar y la cantidad. Si el cliente no ha reservado nunca con ese DNI se le creará una cuenta para poder gestionar sus reservas.

AR-RF-02 - CRUD de Apartamentos: Cada apartamento debe tener una página en la que se incluyan fotos, texto, título y camas disponibles por habitación.

AR-RF-03 - CRD Tarifas: El usuario debe poder crear nuevas tarifas con un precio estándar por noche tanto de día de diario como en fin de semana.

AR-RF-04 - CRD Periodos: El usuario debe poder incluir periodos a una tarifa en los que el precio subirá o bajará dependiendo de la demanda, de esta manera subiendo el precio en periodos como navidad o rebajando los precios en otros meses menos demandados.

AR-RF-05 - CRD Anticipos: El usuario debe poder crear un nuevo periodo de anticipo introduciendo una fecha de inicio y otra final, de manera que si un cliente quiere reservar dentro de ese periodo, se le notificará durante el proceso que deberá ingresar en cuenta un 10% del precio total si quiere realizarla.

AR-RF-06 - Consultar Reservas: El cliente podrá consultar las reservas que ha realizado en la aplicación y el usuario podrá consultar las reservas de todos los clientes, pudiendo además filtrarlas introduciendo un DNI de cliente.

AR-RF-07 - Validación de Reservas: El usuario deberá confirmar las peticiones de reserva que han hecho los clientes, cambiando el estado de la reserva.

AR-RF-08 - Valoración de Reservas: Cuando el usuario consulta las reservas puede valorarlas en función del estado en el que el cliente ha abandonado los apartamentos.

AR-RF-09 - Cumplimentar encuesta de satisfacción: Los huéspedes, una vez terminada la reserva, tendrán la opción de realizar un cuestionario de satisfacción.

AR-RF-10 - Consultar encuestas de satisfacción: El usuario podrá visualizar las encuestas de satisfacción rellenas por los clientes desde su panel de control.

AR-RF-13 - CRUD Oferta: El usuario podrá crear una oferta, modificar la oferta existente o desactivarla. En un primer momento el usuario solo quiere mostrar una única oferta que va modificándose y aparecerá un botón visible para los clientes en la página principal en caso de que esté activada.

AR-RF-14 - CRUD Clientes: Cuando los clientes realicen una petición de reserva el sistema les creará una cuenta con su correo electrónico y contraseña introducidas, de manera que podrán iniciar sesión en la página web, acceder a sus reservas y completar cuestionarios de satisfacción.

AR-RF-15- Login: Los clientes con una cuenta deben poder conectarse como usuarios para ver sus reservas.

AR-RF-16: Seguir Conectado: La página deberá implementar un token para mantenerse conectado durante 30 días.

3.2 Requisitos No Funcionales

AR-RNF-01 - Seguridad: Dentro de la web se encuentran tanto las secciones para usuarios normales como el panel de administración. Debe controlarse que los usuarios no puedan acceder a este si no tienen el rol de administrador.

AR-RNF-02 - Tolerancia a fallos: La web debe estar preparada para que no haya errores, como por ejemplo introducir una fecha de reserva anterior a la fecha actual.

AR-RNF-03 - Tiempo de respuesta: La web debe estar preparada para que la base de datos realice las consultas en un tiempo razonable. Una página no debe superar el tiempo de carga de 4 segundos.

AR-RNF-04 - Mantenibilidad: La web debe estar preparada para que, cuando esté finalizada, un usuario con rol de administrador pueda mantener la web sin necesidad de que el programador vuelva a tocar el código.

AR-RNF-05 - Capacidad de Carga: El sistema debe estar dimensionado para trabajar como mínimo con 8 apartamentos, 10.000 reservas y 5000 usuarios.

AR-RNF-06 - Compatibilidad Software: El sistema debe estar preparado para poder visualizarse, al menos, en la versión de escritorio de los navegadores Google Chrome, Mozilla Firefox y Microsoft Edge. Además, debe estar preparado para que en un futuro se pueda implementar responsive y hacerlo visible en smartphones fácilmente.

3.3 Requisitos de Información

AR-RI-01 - Usuarios: DNI/NIE/Pasaporte, nombre y apellidos, e-mail, teléfono, fecha de nacimiento y rol dentro de la web.

AR-RI-02 - Reservas: Fecha de entrada, fecha de salida, usuario que la ha realizado, número de adultos, número de menores, estado, valoración, apartamentos implicados, precio y cantidad de anticipo.

AR-RI-03 - Apartamentos: Imágenes, nombre, descripción larga, descripción corta.

AR-RI-04 - Oferta: Título, texto y opción de activación/desactivación.

AR-RI-05 - Tarifas: Precio de día de diario, precio de fin de semana, número de noches y apartamento al que están asociadas.

AR-RI-06 - Periodos: Fecha de inicio, fecha de fin, cantidad de suma y tarifa asociada.

AR-RI-07 - Ocupados: Fecha en el que se encuentra ocupado y el apartamento asociado.

AR-RI-08 - Habitaciones: Nombre, número de camas de matrimonio, número de camas individuales, número de sofás cama y el apartamento al que están asociadas.

AR-RI-08 - Cuestionarios: Calidad, limpieza, comentario y reserva asociada al cuestionario.

AR-RI-09 - Anticipos: Fecha de inicio, Fecha de fin.

3.4 Diagrama de Casos de Uso

Una vez hemos sacado los requisitos, nos sentamos con el cliente y definimos los roles de usuarios que hay: Usuario no conectado, Usuario conectado y Administrador.

Seguidamente, acordamos con el cliente que puede hacer cada uno de ellos, asignando los requisitos funcionales y generando un diagrama de casos de uso con Magicdraw. Esto nos permitirá ver de forma visual el funcionamiento del sistema y detectar algún posible error con el cliente.

Los roles que se conectan mediante flecha a otros roles heredan los casos de uso de los mismos, siendo así; el administrador puede hacer lo que hace un usuario conectado, y un usuario conectado puede hacer lo que hace un usuario no conectado.

Además, comentar que si un caso de uso extiende de otro, esto significa que para realizar un caso de uso es necesario realizar el anterior previamente.



Figura 12: Diagrama de Casos de Uso UML

4

Análisis y Diseño

Una vez obtenidos los requisitos procederemos a analizar y diseñar el sistema a implementar. Para ello, diseñaremos una maqueta de la interfaz de usuario en la plataforma Balsamiq que nos permitirá comprobar si estamos captando la idea acerca de qué necesita nuestro cliente. Si algo se nos pasa por alto, será fácil que el cliente lo identifique gracias a este maquetado.

Seguidamente, procederemos con el modelado UML, que es muy importante ya que nos permitirá especificar, visualizar y documentar todo el sistema completo. Una vez realizamos el modelado UML, el sistema queda totalmente definido, lo cual nos facilitará muchísimo la implementación y, además, quedará documentado para poder visualizarlo en cualquier momento.

4.1 Maquetación en Balsamiq

Balsamiq Mockups es una herramienta bastante elaborada que permite a cualquier usuario crear borradores de aplicaciones web, tanto para visión de sobremesa como visión smartphone. El sistema incluye botones, contenedores, imágenes, scroll, formularios, imágenes y todo lo necesario para definir vistas y enlazarlas entre ellas linkeando botones a otras vistas (Figura 13)

Aunque no esté desarrollado de forma funcional, permite modelar una simulación para mostrar al cliente una versión aproximada de lo que sería el sistema antes de comenzar a implementar.



Figura 13: Elementos de maquetación en Balsamiq

Dicho esto, procederemos a visualizar algunas de las vistas de la web de apartamentos. En primer lugar, vamos a ver cómo veríamos el primer módulo descrito en el punto anterior, la página principal y el sistema de reserva de apartamentos (Figuras 14, 15 y 16).

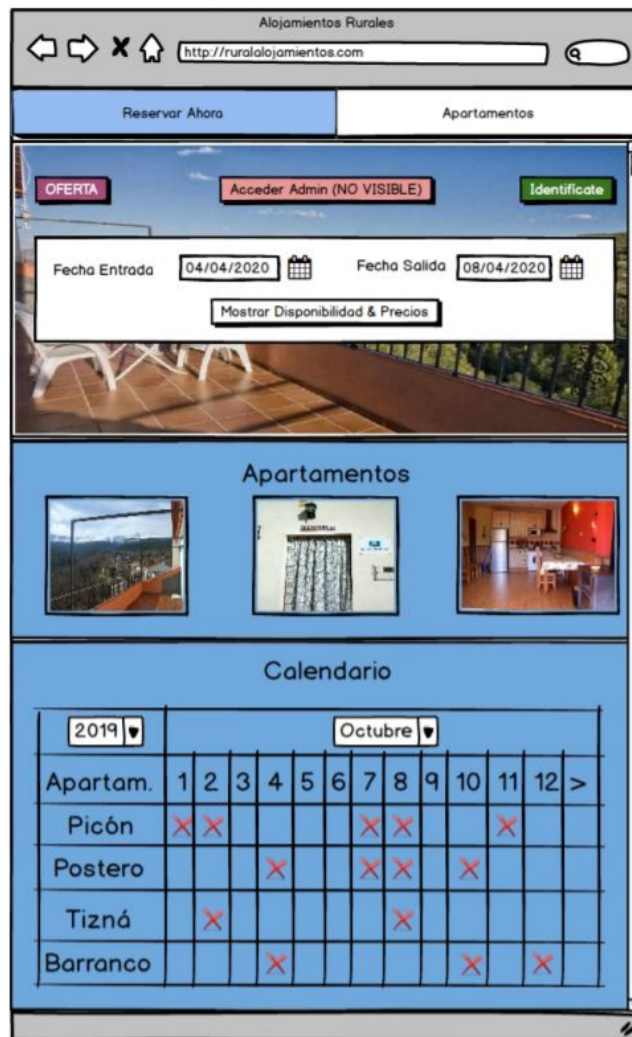


Figura 14: Maqueta de la página principal en Balsamiq



Figura 15: Maqueta para seleccionar apartamentos de reserva en Balsamiq

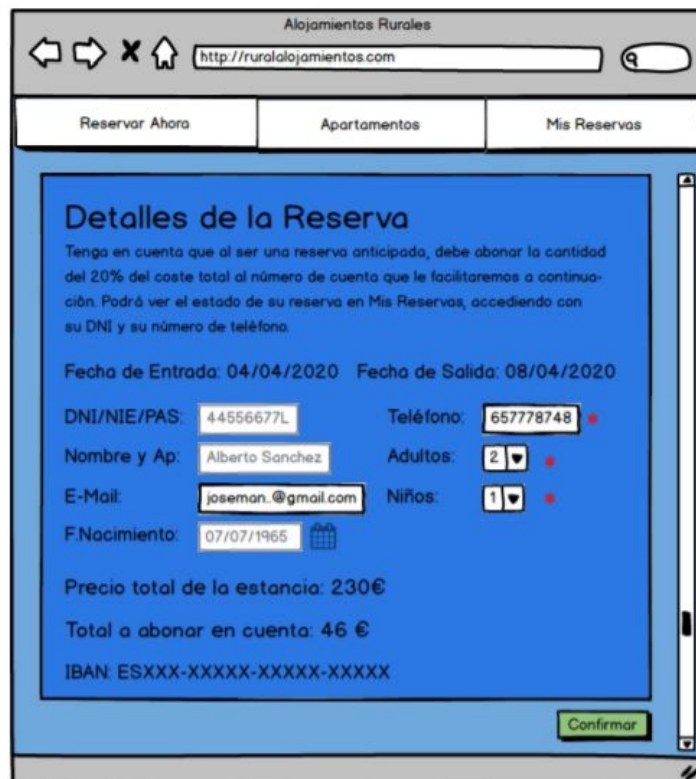


Figura 16: Maqueta de confirmación de reserva en Balsamiq

Vamos a ver ahora algunas de las vistas del segundo módulo, los apartamentos y sus páginas individuales (Figuras 17 y 18).

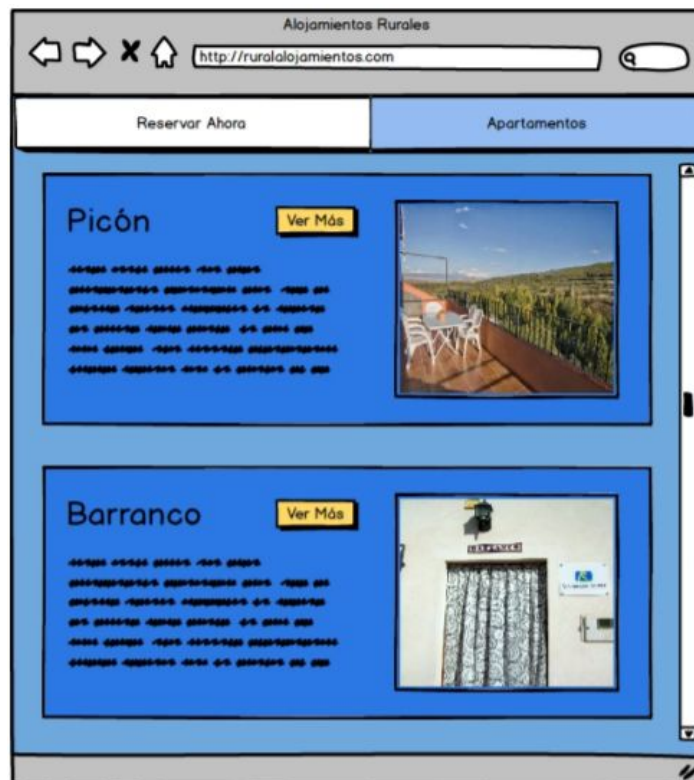


Figura 17: Diapositiva de vista general de apartamentos en Balsamiq

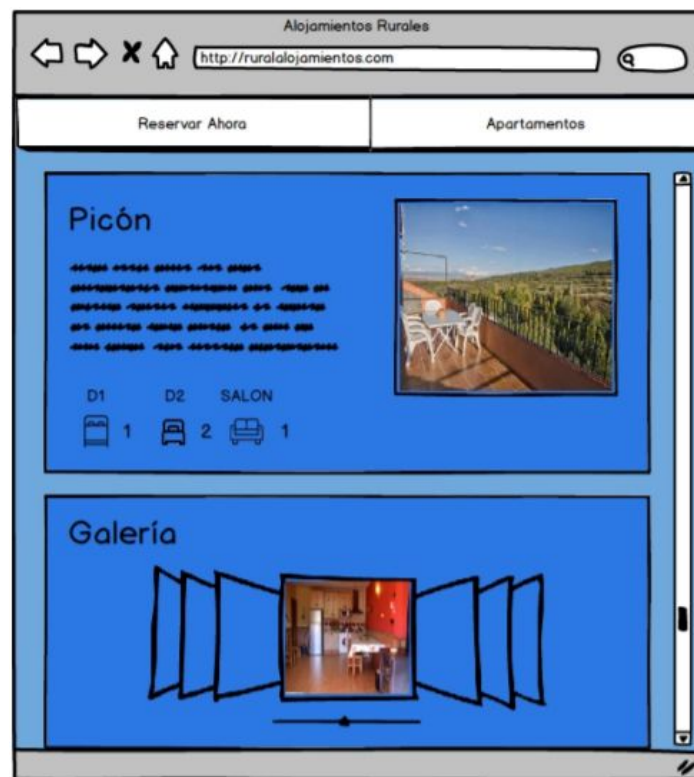


Figura 18: Diapositiva de vista específica de apartamentos en Balsamiq

Vamos a ver ahora algunas vistas del tercer módulo, las reservas de los clientes y el formulario de los cuestionarios de satisfacción (Figuras 19, 20 y 21).

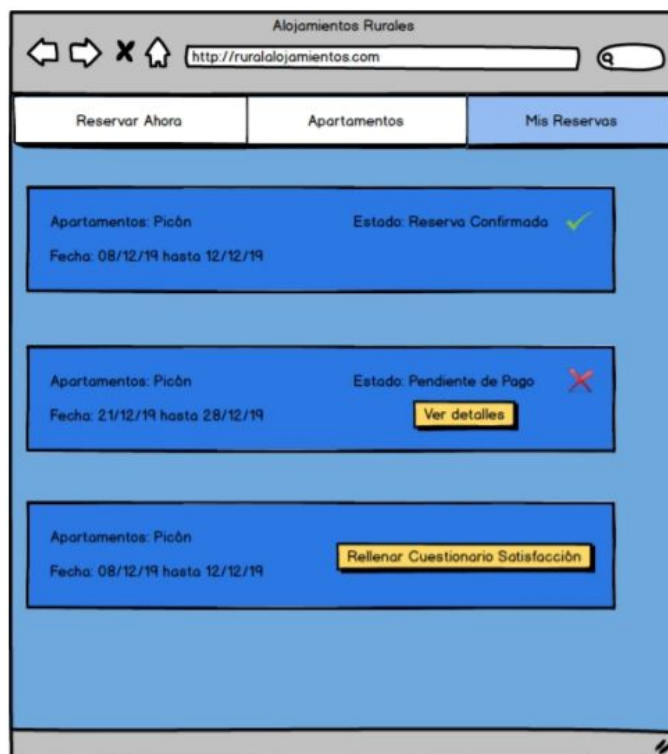


Figura 19: Diapositiva de gestión de Mis Reservas en Balsamiq

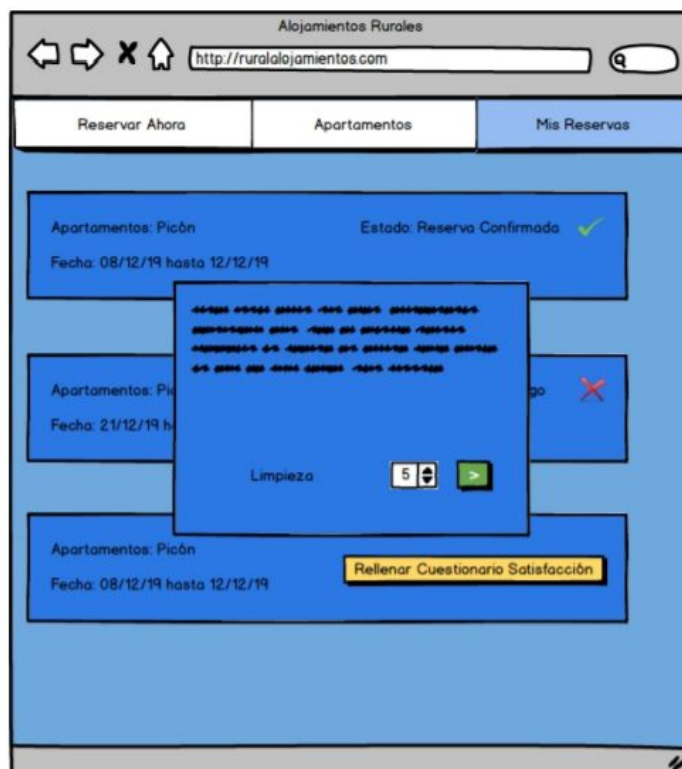


Figura 20: Diapositiva de formulario de satisfacción en Balsamiq

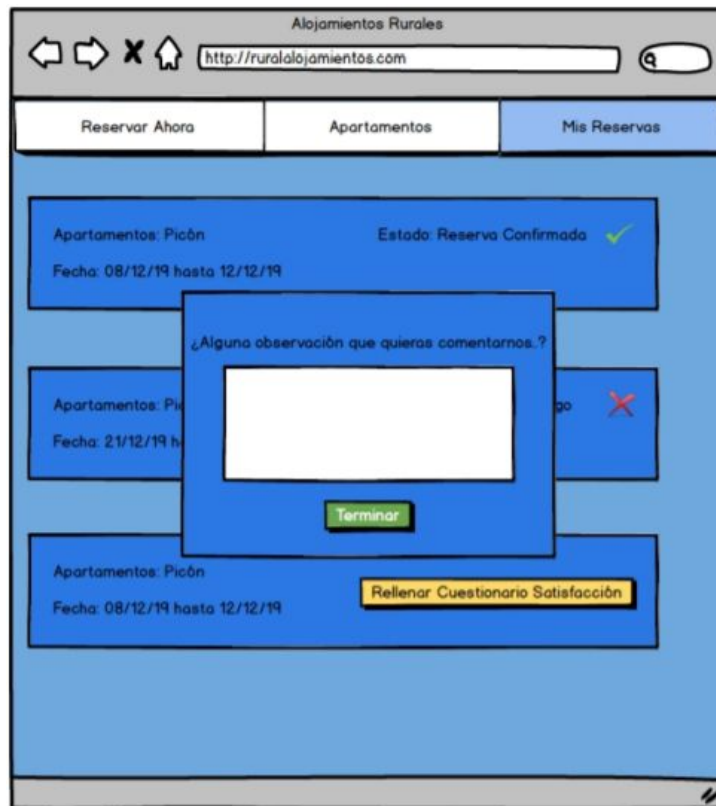


Figura 21: Diapositiva de formulario de satisfacción en Balsamiq

Por último, veremos algunas vistas del cuarto módulo, el panel de administrador (Figuras 22, 23, 24, 25 y 26).

Alojamientos Rurales
http://ruralalojamientos.com/administrador

Tarifas & Disponibilidad Reservas Apartamentos Cuestionarios Oferta y Anticipos

Apartamento: **Picón** Cerrar Sesión

2019 **Octubre**

Tarifas / Días	1	2	3	4	5	6	7	8	9	10	11	12	>
1N	15	15	15	15	18	18	18	15	15	15	15	15	
2N	18	18	18	18	22	22	22	18	18	18	18	18	
Ocupados		X		X	X					X	X		

Crear Tarifa:

Noches: **3** Precio por noche: **23** Añadir

Eliminar Tarifa:

Noches: **2** Eliminar

Figura 22: Diapositiva de gestión de tarifas y disponibilidad en Balsamiq

Alojamientos Rurales
http://ruralalojamientos.com/administrador

Tarifas & Disponibilidad Reservas Apartamentos Cuestionarios Oferta y Anticipos

DNI/PASS: **44678874L** Filtrar Añadir Reserva Añadir Cliente

Apartamentos: **Picón** Estado: **Pendiente de Pago**

Fecha: 08/12/19 hasta 12/12/19 Valoración: **Seleccionar**

Nombre: Alberto Sánchez
DNI: 44568874L
Telefono: 666555666
E-Mail: albertorg@uma.es
Nacimiento: 07/01/1965

Adultos: 2
Menores: 1

Guardar Cambios

Apartamentos: **Tizná** Estado: **Pendiente de Pago**

Fecha: 08/12/19 hasta 12/12/19 Valoración: **Seleccionar**

Nombre: María Figueroa
DNI: 36674892M
Telefono: 603455678
E-Mail: mariaf@yahoo.com

Adultos: 2
Menores: 1

Figura 23: Diapositiva de gestión de reservas en Balsamiq



Figura 24: Diapositiva de gestión de apartamentos en Balsamiq

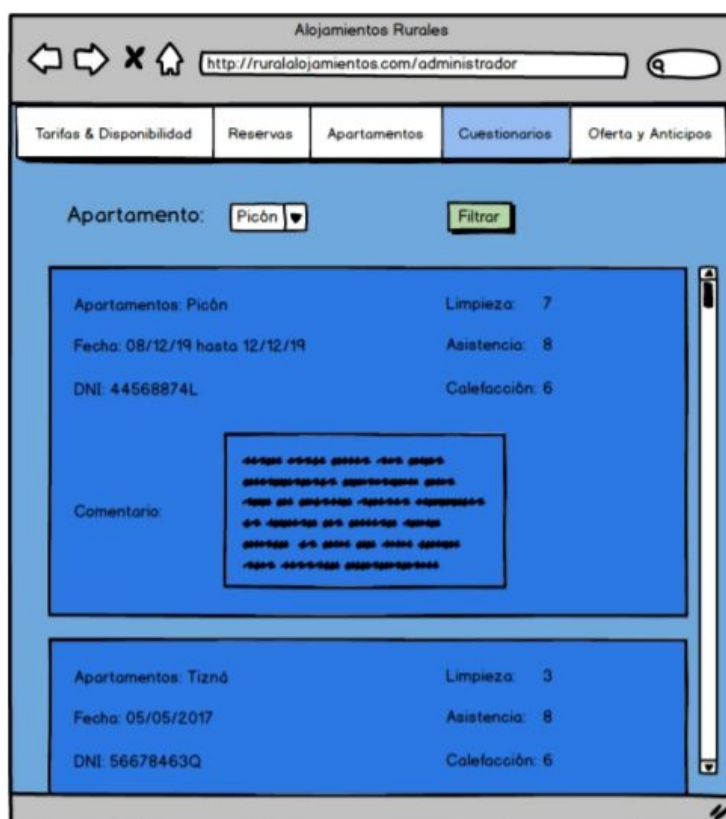


Figura 25: Diapositiva de visualización de cuestionarios en Balsamiq

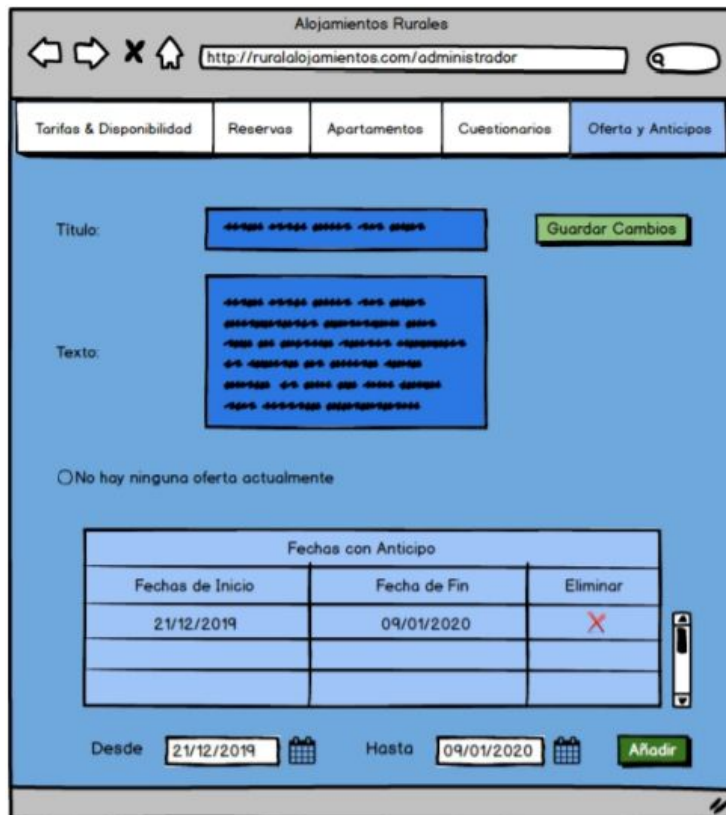


Figura 26: Diapositiva de gestión de oferta y anticipos en Balsamiq

Tras haber enseñado la maqueta al usuario dueño de la web, ha quedado bastante satisfecho con la maqueta y, por lo pronto no nos pide nuevas funcionalidades extra a las que aparentemente ha visto en este modelado. Por lo tanto, procedemos a definir el sistema de una manera más concreta con el modelado UML.

4.2 Modelado UML

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

MagicDraw es una herramienta de modelado que permite una representación visual del proyecto. Diseñado para los analistas de negocio, analistas de software, programadores e ingenieros de control de calidad, esta herramienta de desarrollo dinámico y versátil facilita el análisis y diseño de la orientación a objetos, sistemas y bases de datos.

El primer diagrama a realizar es el diagrama de clases de entidad (Figura 27), que nos va a dar las entidades que posteriormente se generarán en la base de datos. Para ello nos fijamos en los Requisitos de Información, ya que por lo general son muy parecidos a las entidades.

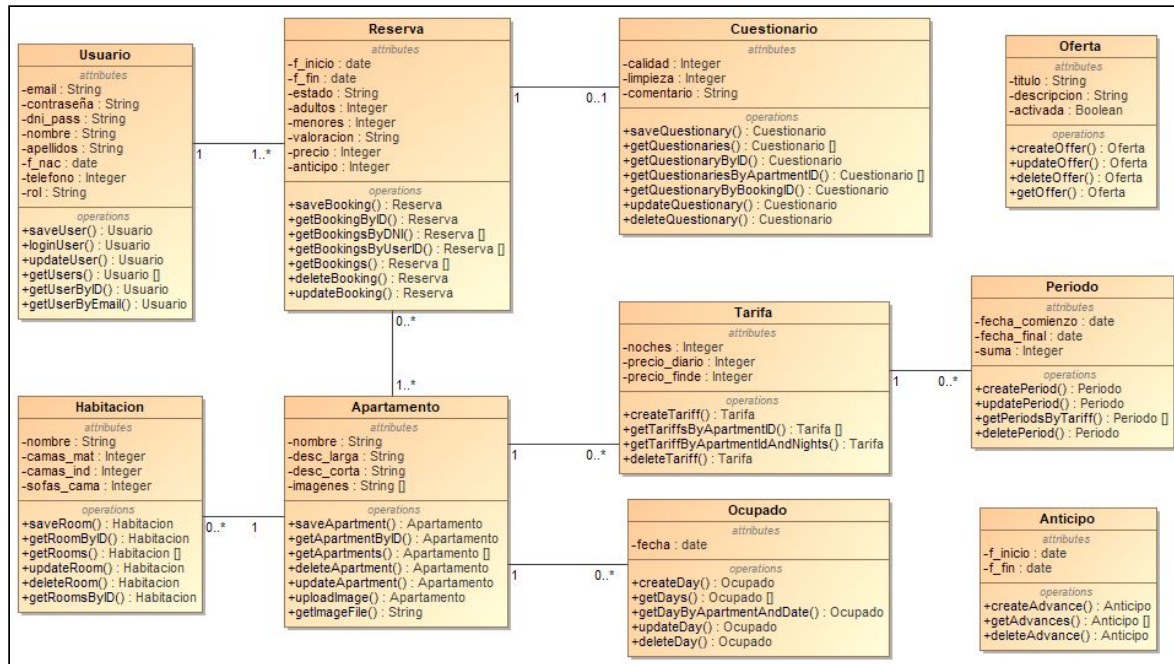


Figura 27: Diagrama de Clases en MagicDraw

Además, los métodos de las clases nos darán los futuros métodos a implementar en los controladores de la base de datos para construir la futura ‘API Rest’.

Una vez diseñado el diagrama de clases proseguimos con el Modelo Físico de Datos (Figura 28), que es muy parecido al Diagrama de Clases pero especificando las claves foráneas que serán necesarias para enlazar las entidades, dependiendo del tipo de relación que haya entre ellas.

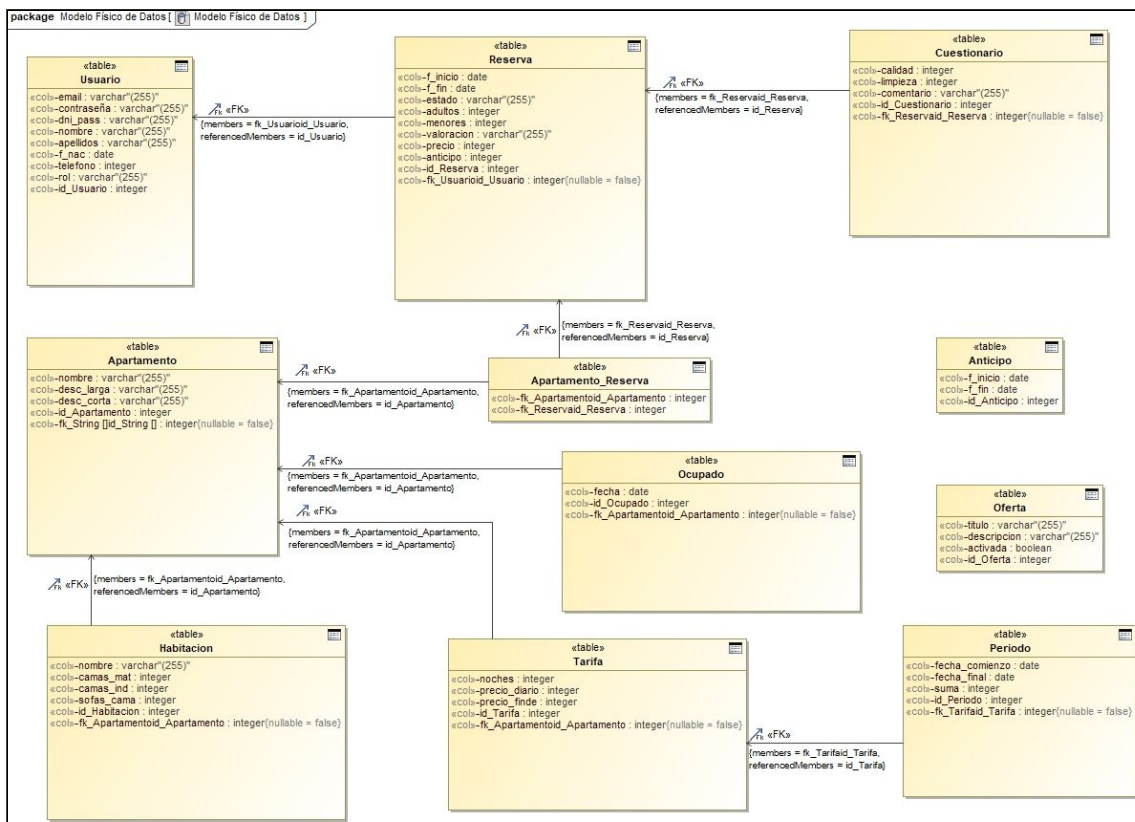


Figura 28: Modelo Físico de Datos en MagicDraw

Este diagrama es fundamental para la generación de la base de datos de la aplicación, puesto que nos ayudará a crear los objetos en el desarrollo de la API con sus claves foráneas correspondientes.

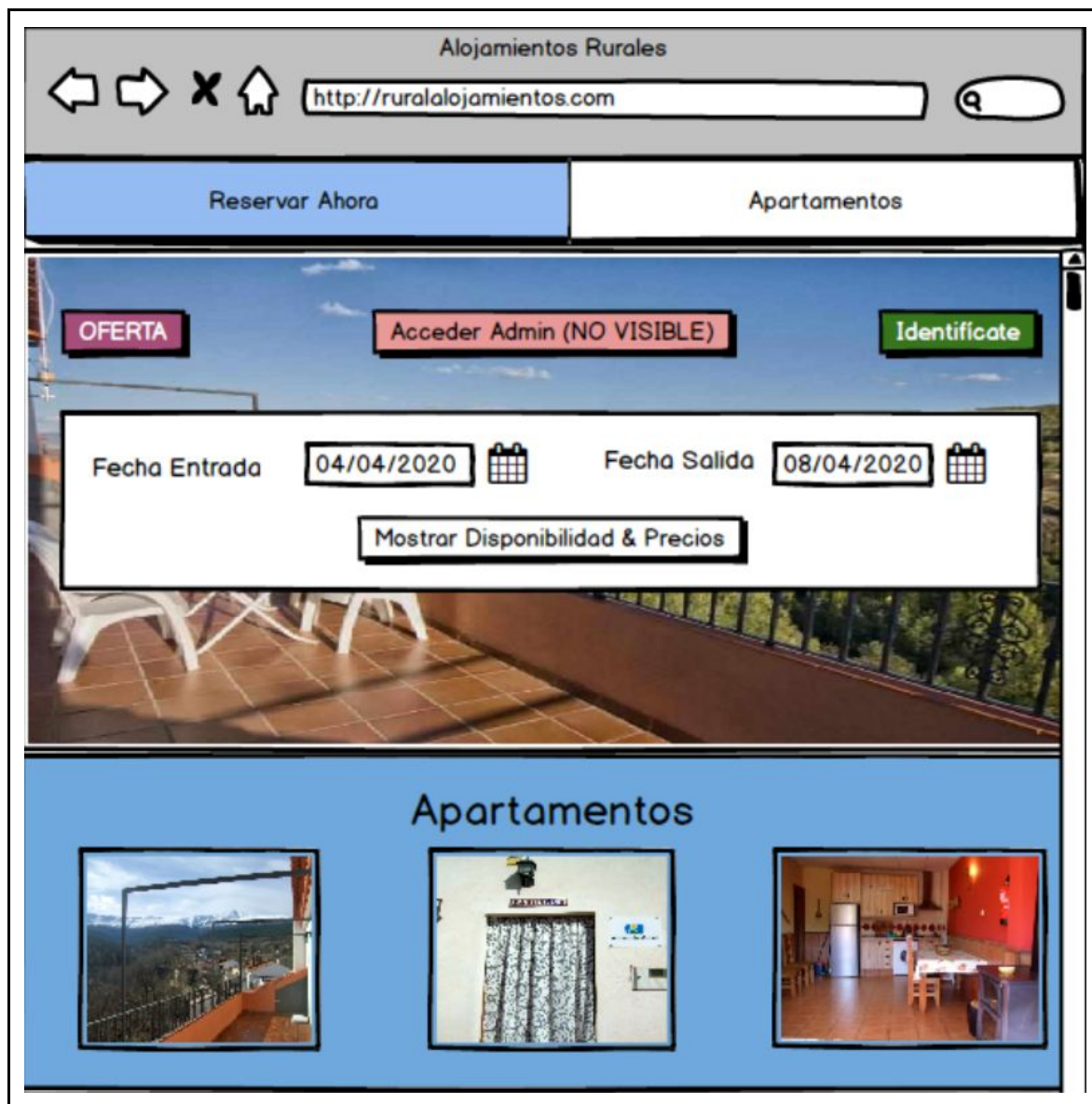
4.3 Especificación de Casos de Uso

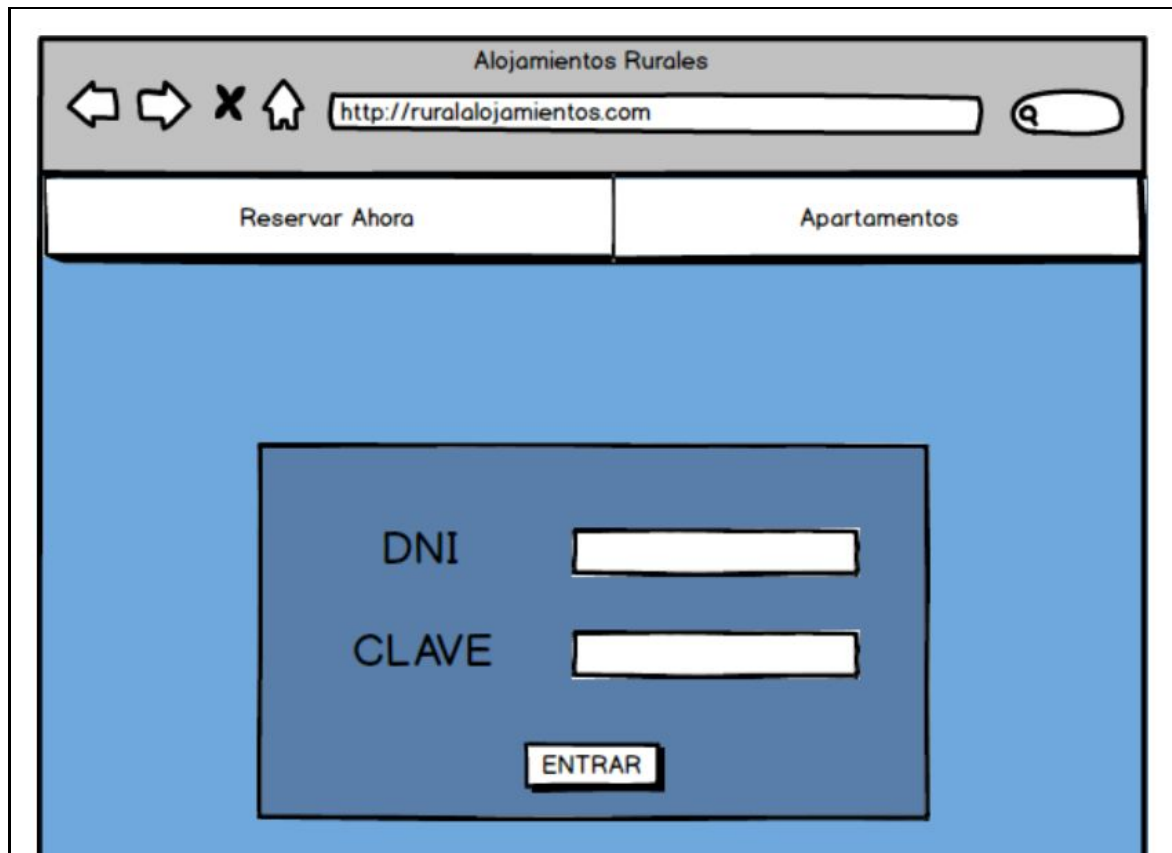
Una vez realizado todo el modelado, podemos proceder a especificar el diagrama de casos de uso que realizamos en el punto 2.2. Gracias a los diagramas de secuencia, esta especificación nos permitirá conocer mucho mejor los procesos y definirlos de forma que se nos haga muy sencilla la implementación del código, ya que incluso saldrán la mayoría de métodos que debemos implementar en los componentes.

4.3.1 Log-in

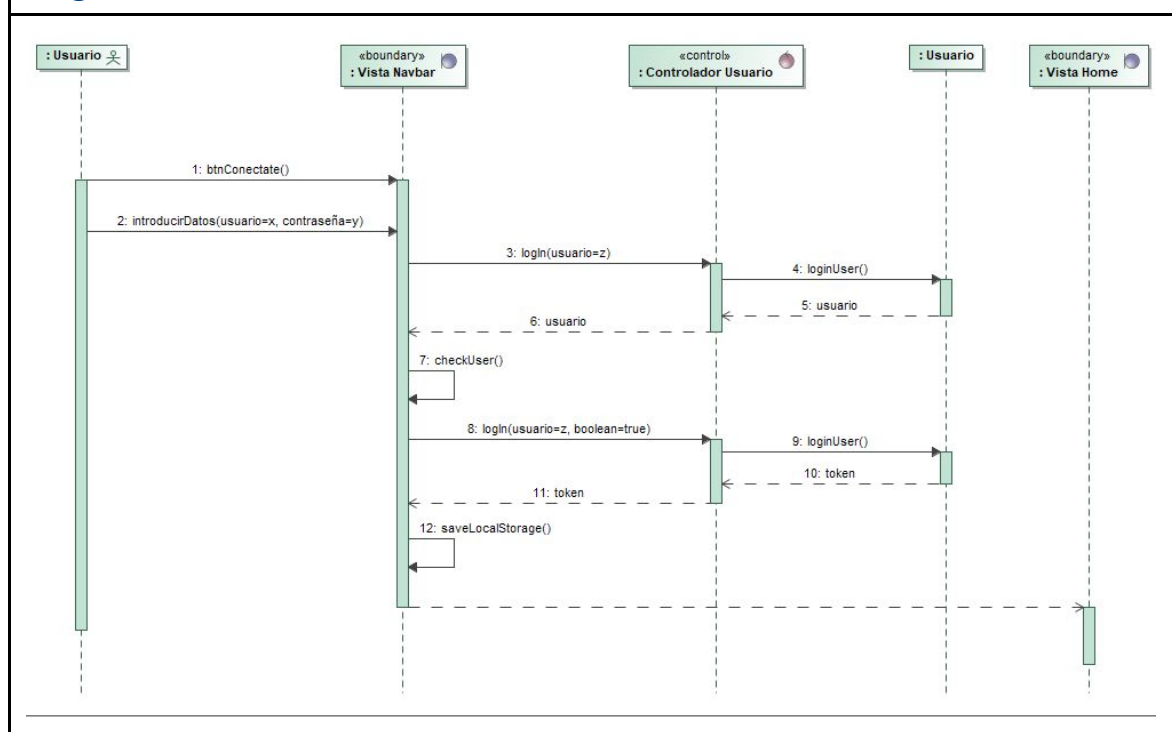
Título	Log-In
Descripción	Logearse como usuario en la aplicación web
Pre-condición	No estar conectado y estar registrado como usuario en la aplicación. Debe existir un usuario con nombre y contraseña en la base de datos

Post-condición	El usuario se ha conectado, puede acceder a sus reservas y al panel de administrador si posee el rol necesario. Se crea además un token en el local storage que dura 30 días	
Prioridad	Alta	
Autor(es)	José María Gómez García	
Control de cambios	Última modificación: 7 de Septiembre de 2020	
Escenario principal		
1. El usuario accede a la web y presiona el botón de Conéctate en la barra de navegación 2. Se abre un desplegable en el que el usuario introduce su usuario y contraseña 3. El usuario se conecta y puede acceder a nuevas secciones en la web		
Escenario alternativo		
1. El usuario accede a la web y presiona el botón de Conéctate en la barra de navegación 2. Se abre un desplegable en el que el usuario introduce su usuario y contraseña 3. Los datos son incorrectos y el sistema le avisa de que las credenciales son erróneas		
Clases de análisis		
A. Clases de entidad	Usuario	
B. Clases de control	navbar.component.ts	
C. Clases de interfaz	navbar.component.html	
Maquetas de interfaz		





Diagramas de secuencia



4.3.2 Realizar Reserva

Título	Realizar Reserva
Descripción	Un usuario realiza una reserva de apartamentos
Pre-condición	Ninguna precondition, puesto que cualquier tipo de usuario puede realizar la reserva, incluso si no está registrado
Post-condición	Se ha registrado la reserva en la base de datos, y, en caso de haber sido un usuario no registrado el que la ha realizado, también se crea un usuario y se asocia la reserva a dicho usuario. Además, se crean nuevos días ocupados para las fechas y apartamentos correspondientes, de forma que se actualiza el calendario de disponibilidad
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
<ol style="list-style-type: none">1. El usuario conectado accede a la web y presiona el botón de Inicio de la barra de navegación2. El usuario introduce las fechas en las que desea alojarse y presiona el botón ‘Mostrar Disponibilidad & Precios’3. El sistema le ofrece los apartamentos que puede seleccionar para esas fechas, escoge los que quiere alquilar y presiona en el botón ‘Reservar’.4. El usuario introduce los datos restantes para completar la reserva y presiona el botón ‘Confirmar Reserva’5. La reserva se ha realizado correctamente y el usuario puede visualizar los datos de su reserva en el apartado ‘Mis Reservas’	
Escenario alternativo	
<ol style="list-style-type: none">1. El usuario conectado accede a la web y presiona el botón de Inicio de la barra de navegación2. El usuario introduce las fechas en las que desea alojarse y presiona el botón ‘Mostrar Disponibilidad & Precios’3. El sistema le ofrece los apartamentos que puede seleccionar para esas fechas, escoge los que quiere alquilar y presiona en el botón ‘Reservar’.4. El usuario introduce los datos restantes para completar la reserva y presiona el botón ‘Confirmar Reserva’5. Algunos datos introducidos no son correctos y el sistema pide al usuario que revise los campos	

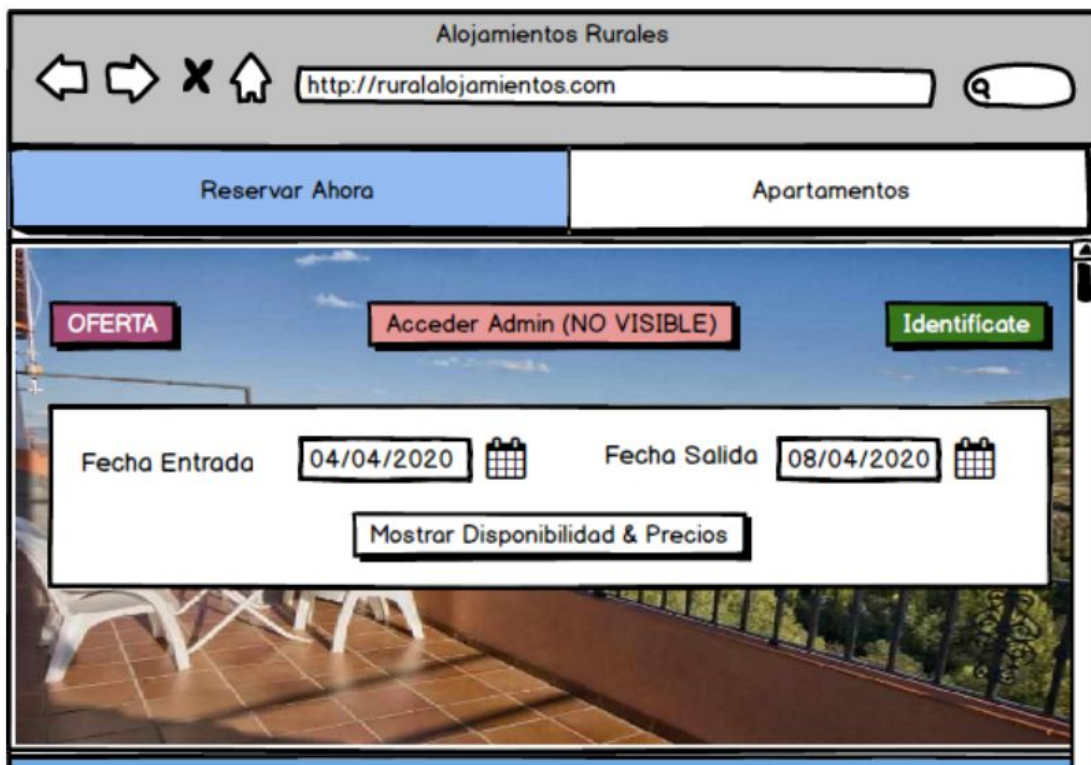
Clases de análisis	
A. Clases de entidad	Reserva, Usuario, Ocupado, Anticipo, Tarifa, Periodo, Apartamento, Habitación
B. Clases de control	inputdates.component.ts, reservar.component.ts
C. Clases de interfaz	inputdates.component.html, reservar.component.html
Maquetas de interfaz	









Reservar Ahora	Apartamentos	Mis Reservas																				
<h2>Detalles de la Reserva</h2> <p>Tenga en cuenta que al ser una reserva anticipada, debe abonar la cantidad del 20% del coste total al número de cuenta que le facilitaremos a continuación. Podrá ver el estado de su reserva en Mis Reservas, accediendo con su DNI y su número de teléfono.</p> <p>Fecha de Entrada: 04/04/2020 Fecha de Salida: 08/04/2020</p> <table><tr><td>DNI/NIE/PAS:</td><td><input type="text" value="44556677L"/></td><td>Teléfono:</td><td><input type="text" value="657778748"/></td><td>*</td></tr><tr><td>Nombre y Ap:</td><td><input type="text" value="Alberto Sanchez"/></td><td>Adultos:</td><td><input type="text" value="2"/></td><td>▼ *</td></tr><tr><td>E-Mail:</td><td><input type="text" value="joseman..@gmail.com"/></td><td>Niños:</td><td><input type="text" value="1"/></td><td>▼ *</td></tr><tr><td>F.Nacimiento:</td><td><input type="text" value="07/07/1965"/></td><td></td><td></td><td></td></tr></table> <p>Precio total de la estancia: 230€</p> <p>Total a abonar en cuenta: 46 €</p> <p>IBAN: ESXXX-XXXXX-XXXXX-XXXXX</p> <p><input type="button" value="Confirmar"/></p>			DNI/NIE/PAS:	<input type="text" value="44556677L"/>	Teléfono:	<input type="text" value="657778748"/>	*	Nombre y Ap:	<input type="text" value="Alberto Sanchez"/>	Adultos:	<input type="text" value="2"/>	▼ *	E-Mail:	<input type="text" value="joseman..@gmail.com"/>	Niños:	<input type="text" value="1"/>	▼ *	F.Nacimiento:	<input type="text" value="07/07/1965"/>			
DNI/NIE/PAS:	<input type="text" value="44556677L"/>	Teléfono:	<input type="text" value="657778748"/>	*																		
Nombre y Ap:	<input type="text" value="Alberto Sanchez"/>	Adultos:	<input type="text" value="2"/>	▼ *																		
E-Mail:	<input type="text" value="joseman..@gmail.com"/>	Niños:	<input type="text" value="1"/>	▼ *																		
F.Nacimiento:	<input type="text" value="07/07/1965"/>																					



Diagramas de secuencia

<p>3. El sistema le ofrece los apartamentos que puede seleccionar para esas fechas, escoge los que quiere alquilar y presiona en el botón ‘Reservar’.</p> <p>4. El usuario introduce los datos restantes para completar la reserva y presiona el botón ‘Confirmar Reserva’</p> <p>5. La reserva se ha realizado correctamente y el usuario se ha almacenado en la base de datos mediante un sistema de token que protege la privacidad de su contraseña hasta de los mismos programadores</p>	
Escenario alternativo	
<p>1. El usuario conectado accede a la web y presiona el botón de Inicio de la barra de navegación</p> <p>2. El usuario introduce las fechas en las que desea alojarse y presiona el botón ‘Mostrar Disponibilidad & Precios’</p> <p>3. El sistema le ofrece los apartamentos que puede seleccionar para esas fechas, escoge los que quiere alquilar y presiona en el botón ‘Reservar’.</p> <p>4. El usuario introduce los datos restantes para completar la reserva y presiona el botón ‘Confirmar Reserva’</p> <p>5. El sistema informa de que el correo electrónico introducido no es correcto puesto que no cumple con el formato correspondiente</p>	
Clases de análisis	
A. Clases de entidad	Reserva, Usuario, Ocupado, Anticipo, Tarifa, Periodo, Apartamento, Habitación
B. Clases de control	inputdates.component.ts, reservar.component.ts
C. Clases de interfaz	inputdates.component.html, reservar.component.html
Maquetas de interfaz	

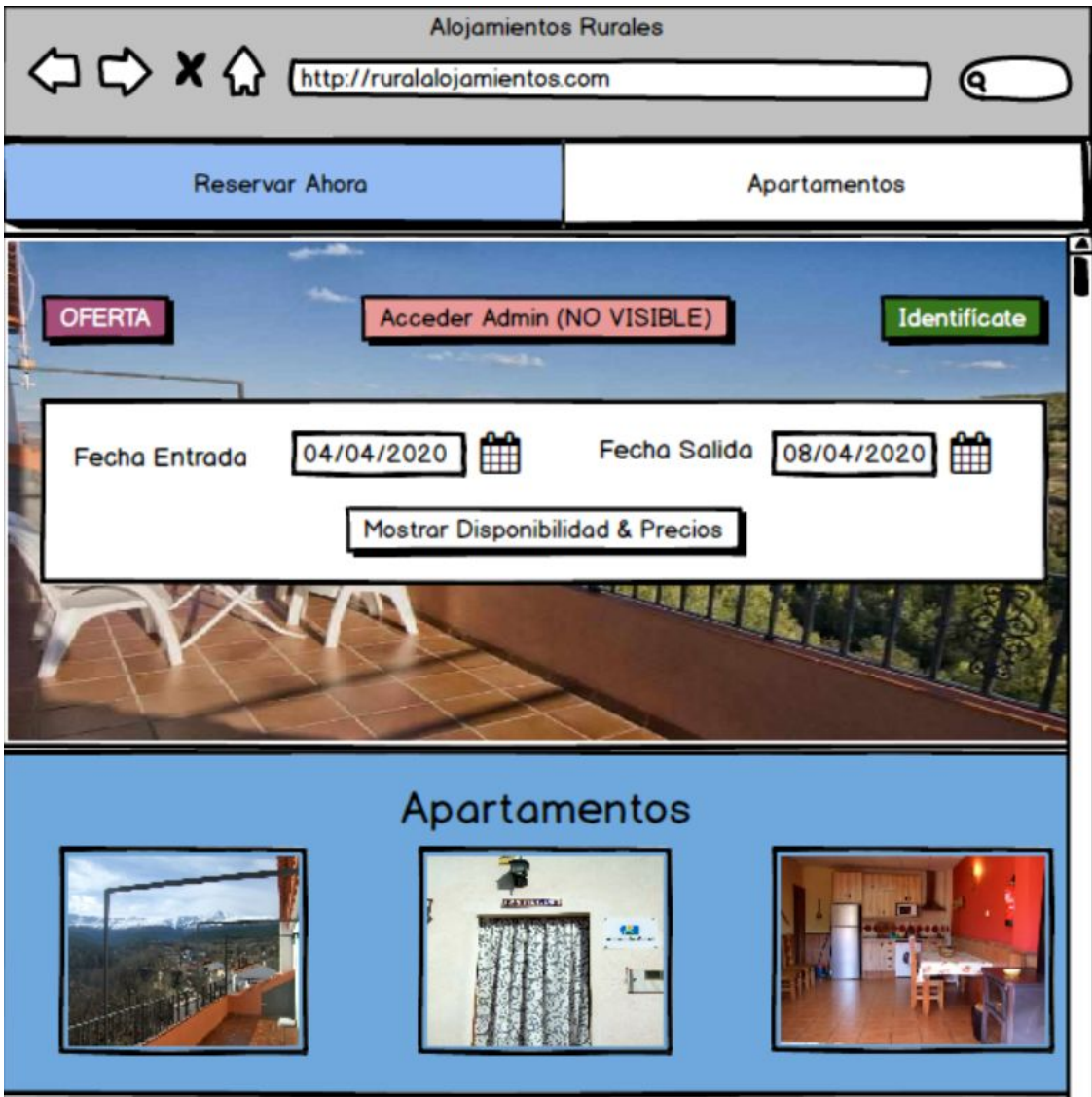


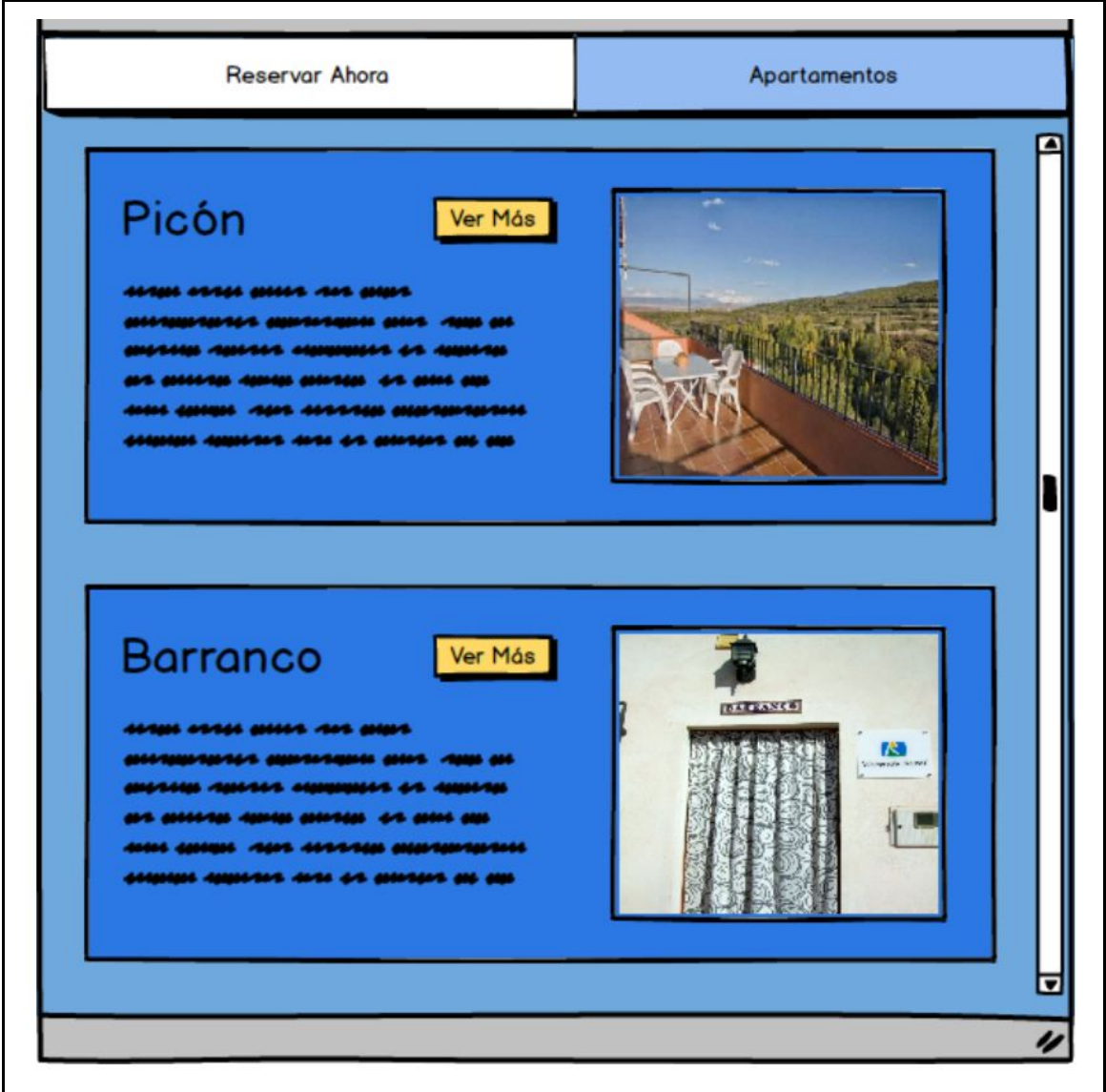
Reservar Ahora	Apartamentos
<div> <div> <h3>Picón</h3> <p> 4+1 personas 1 dormitorio 1 baño 1 cocina 1 sala de estar 1 terraza 1 parking </p> <div>  1  2  1 </div> <div> <input checked="" type="checkbox"/> Seleccionar </div> </div> <div> <div>230€</div>  </div> </div>	
<div> <div> <h3>Postero</h3> <p> 4+1 personas 1 dormitorio 1 baño 1 cocina 1 sala de estar 1 terraza 1 parking </p> <div>  1  2  1 </div> <div> <input type="checkbox"/> Seleccionar </div> </div> <div> <div>210€</div>  </div> </div>	
<div>Realizar Reserva</div>	

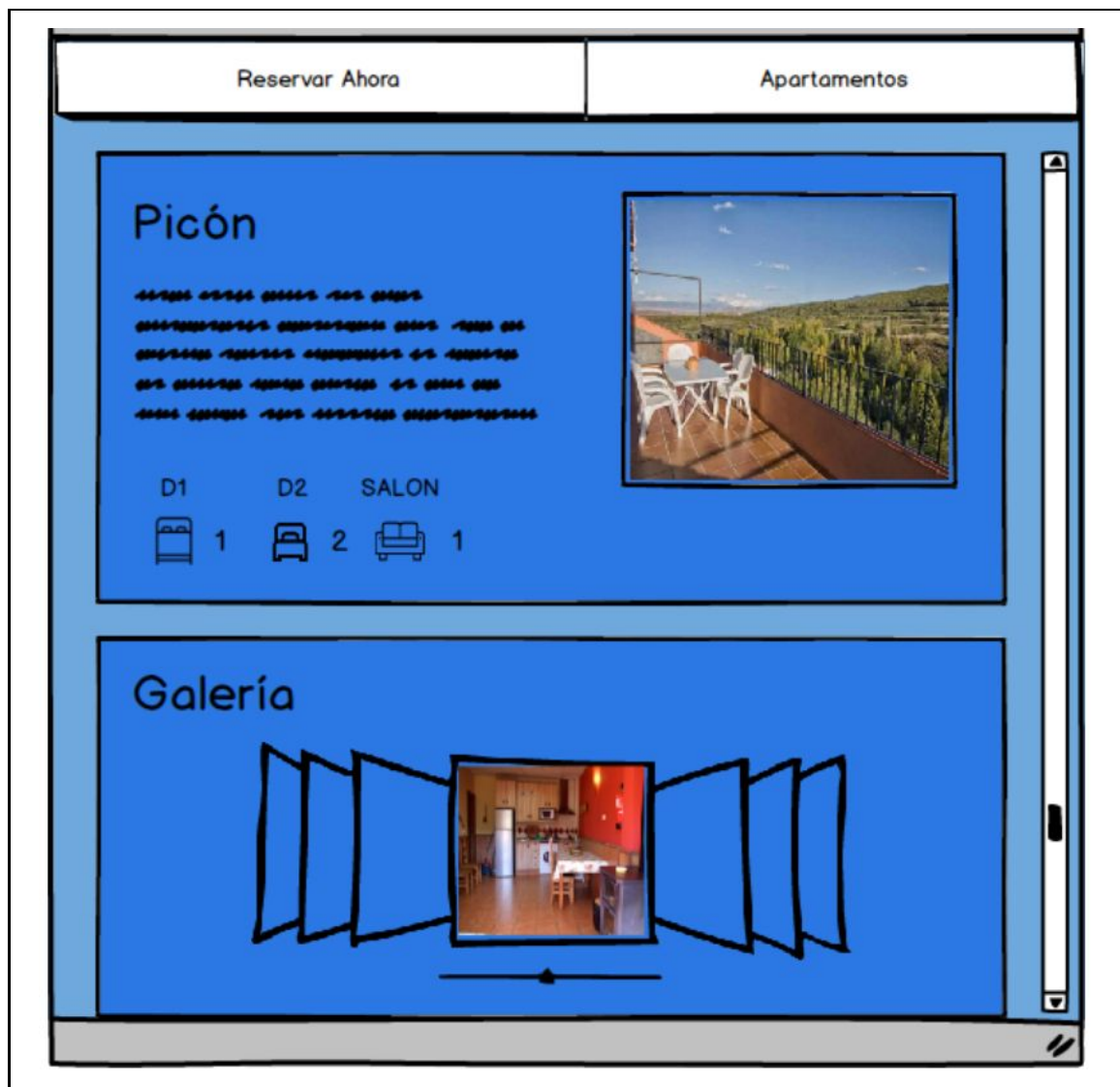
Reservar Ahora	Apartamentos																								
<h2>Detalles de la Reserva</h2> <p>Tenga en cuenta que al ser una reserva anticipada, debe abonar la cantidad del 20% del coste total al número de cuenta que le facilitaremos a continuación. Podrá ver el estado de su reserva en Mis Reservas, accediendo con su DNI y su número de teléfono.</p> <p>Fecha de Entrada: 04/04/2020 Fecha de Salida: 08/04/2020</p> <table><tbody><tr><td>DNI/NIE/PAS:</td><td><input type="text"/></td><td>*</td><td>Teléfono:</td><td><input type="text"/></td><td>*</td></tr><tr><td>Nombre y Ap:</td><td><input type="text"/></td><td>*</td><td>Adultos:</td><td><input type="text" value="2"/></td><td>▼</td></tr><tr><td>E-Mail:</td><td><input type="text"/></td><td>*</td><td>Niños:</td><td><input type="text" value="1"/></td><td>▼</td></tr><tr><td>F.Nacimiento:</td><td><input type="text" value="/ /"/></td><td></td><td colspan="3"></td></tr></tbody></table> <p>Precio total de la estancia: 230€</p> <p>Total a abonar en cuenta: 46 €</p> <p>IBAN: ESXXX-XXXXX-XXXXX-XXXXX</p> <p><input type="button" value="Confirmar"/></p>		DNI/NIE/PAS:	<input type="text"/>	*	Teléfono:	<input type="text"/>	*	Nombre y Ap:	<input type="text"/>	*	Adultos:	<input type="text" value="2"/>	▼	E-Mail:	<input type="text"/>	*	Niños:	<input type="text" value="1"/>	▼	F.Nacimiento:	<input type="text" value="/ /"/>				
DNI/NIE/PAS:	<input type="text"/>	*	Teléfono:	<input type="text"/>	*																				
Nombre y Ap:	<input type="text"/>	*	Adultos:	<input type="text" value="2"/>	▼																				
E-Mail:	<input type="text"/>	*	Niños:	<input type="text" value="1"/>	▼																				
F.Nacimiento:	<input type="text" value="/ /"/>																								

Reservar Ahora	Apartamentos
<h2>Detalles de la Reserva</h2> <p>Tenga en cuenta que al ser una reserva anticipada, debe abonar la cantidad del 20% del coste total al número de cuenta que le facilitaremos a continuación. Podrá ver el estado de su reserva en Mis Reservas, accediendo con su DNI y su número de teléfono.</p> <div><div><div>Reserva realizada con éxito</div><div>Puede ver sus reservas ingresando su DNI en "Identificate"</div><div>Volver a Inicio</div></div><div><div>Fec:</div><div>20</div></div><div><div>DN:</div><div></div></div><div><div>Non:</div><div></div></div><div><div>E-Mail:</div><div><input type="text"/></div><div>*</div></div><div><div>Niños:</div><div><input type="text" value="1"/></div><div>▼</div></div><div><div>F.Nacimiento:</div><div><input type="text" value="/ /"/></div><div></div></div><div><div>Precio total de la estancia:</div><div>230€</div></div><div><div>Total a abonar en cuenta:</div><div>46 €</div></div><div><div>IBAN:</div><div>ESXXX-XXXXX-XXXXX-XXXXX</div></div></div> <div><div>Confirmar</div></div>	

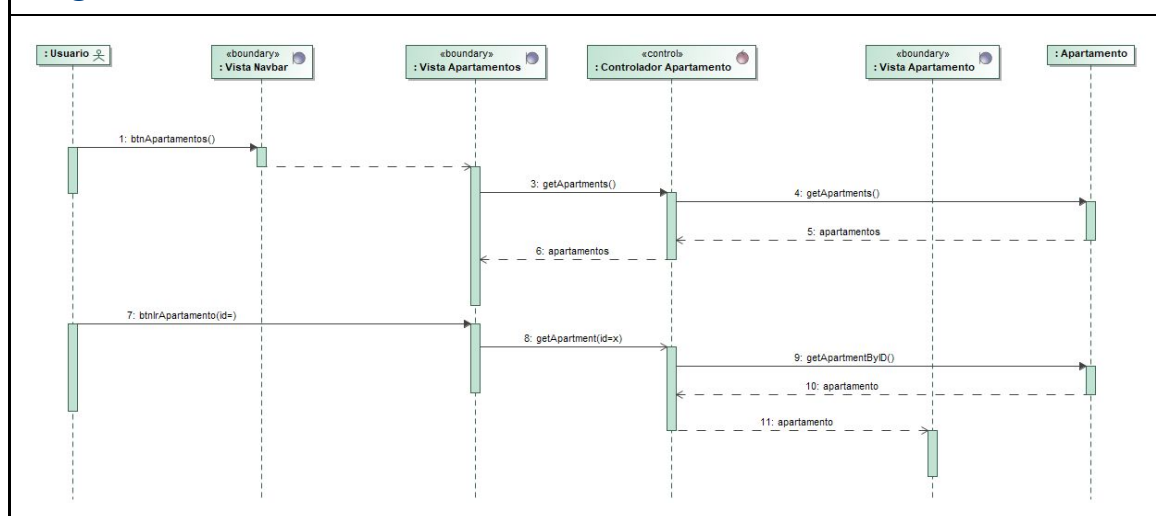
Diagramas de secuencia

Clases de análisis	
A. Clases de entidad	Apartamento, Usuario, Habitación
B. Clases de control	navbar.component.ts, apartamentos.component.ts, apartamento-page.ts
C. Clases de interfaz	navbar.component.html, apartamentos.component.html, apartamento-page.html
Maquetas de interfaz	
	





Diagramas de secuencia



4.3.5 Consultar Calendario de Disponibilidad

Título	Consultar Calendario de Disponibilidad		
Descripción	Visualizar el calendario de disponibilidad de fechas según cada apartamento		
Pre-condición	No existe ninguna precondición		
Post-condición	El usuario puede visualizar qué días están ocupados cada apartamento en un mes y año concretos		
Prioridad	Alta		
Autor(es)	José María Gómez García		
Control de cambios	Última modificación: 7 de Septiembre de 2020		
Escenario principal			
1. El usuario accede a la web y presiona el botón de Inicio en la barra de navegación 2. El usuario introduce el mes y año en el que quiere comprobar la disponibilidad 3. El sistema le muestra qué días está ocupado en dicho mes cada uno de los apartamentos			
Clases de análisis			
A. Clases de entidad		Ocupado, Apartamento	
B. Clases de control		disponibilidad-admin.component.ts	
C. Clases de interfaz		disponibilidad-admin.component.html	
Maquetas de interfaz			

Reservar Ahora

Apartamentos

OFERTA

Acceder Admin (NO VISIBLE)

Identificate

Fecha Entrada




04/04/2020

Fecha Salida

08/04/2020

Mostrar Disponibilidad & Precios

Apartamentos

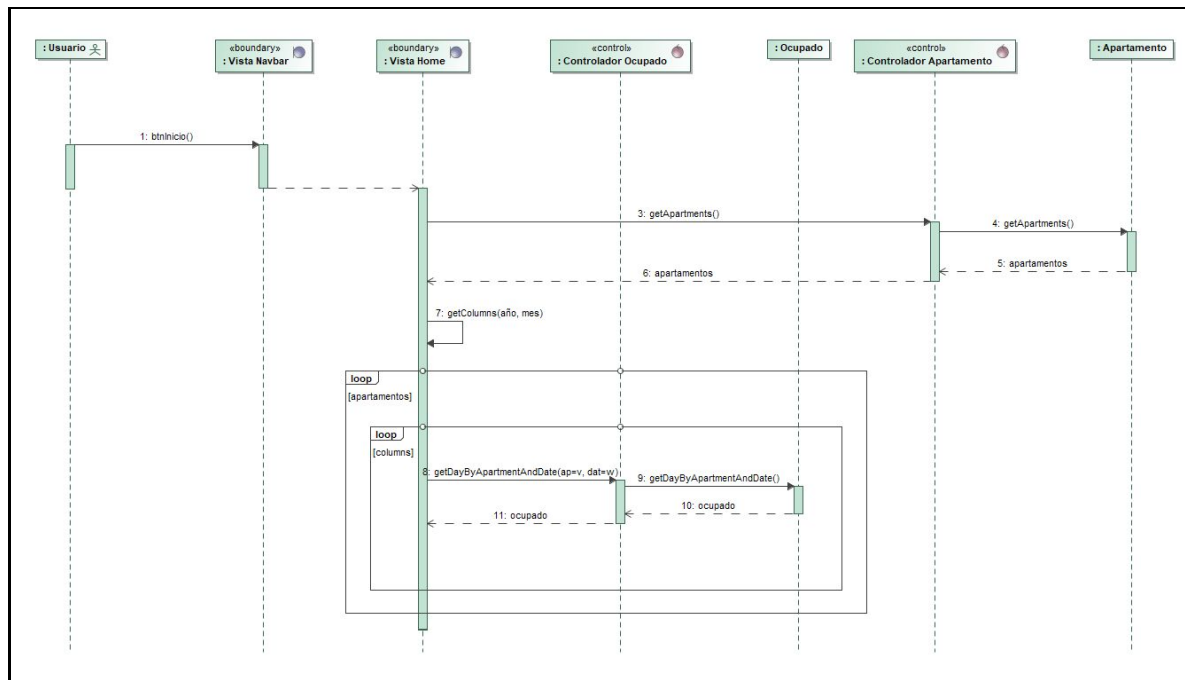




Calendario

2019 ▼

Octubre ▼

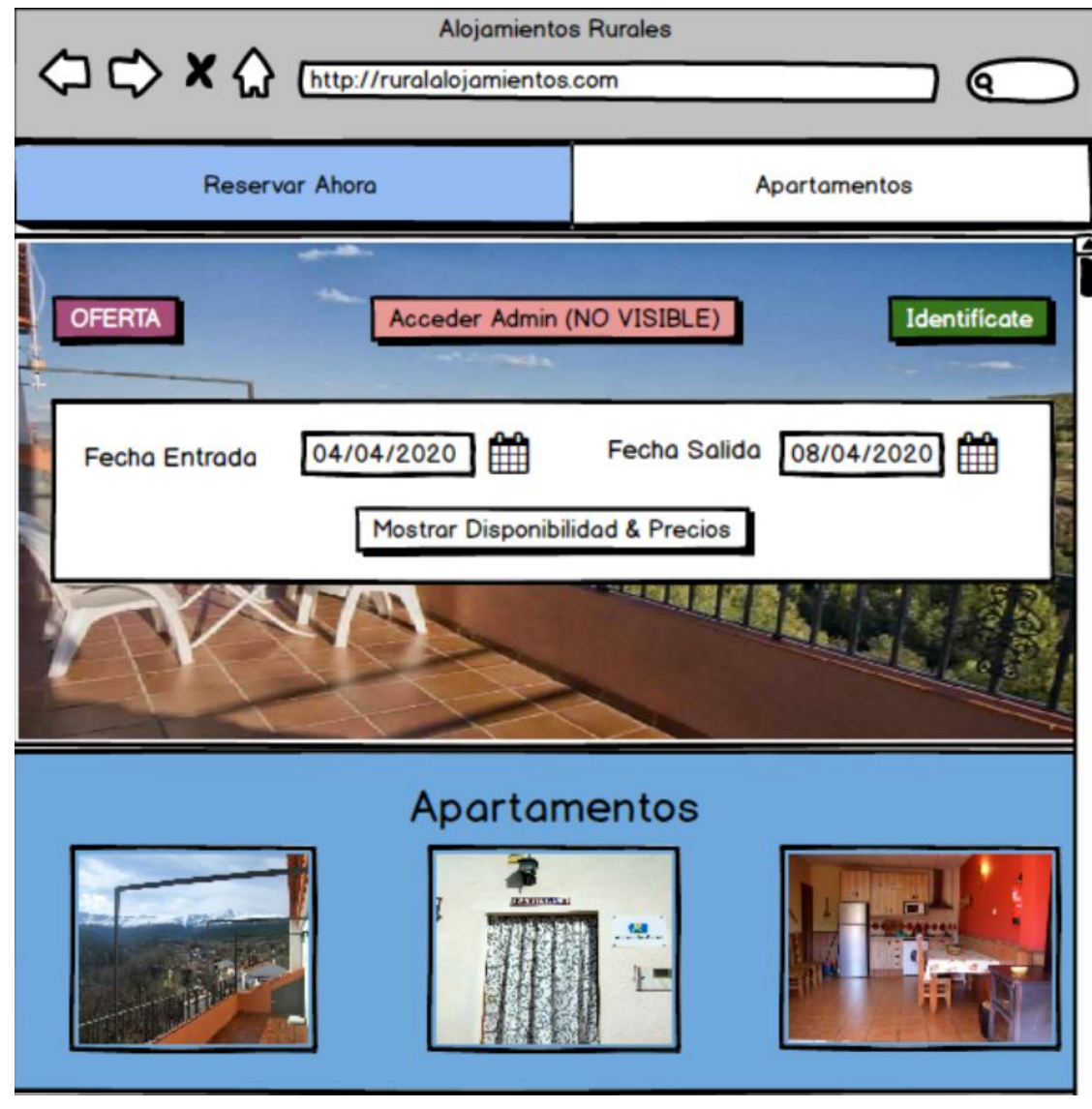
Apartam.	1	2	3	4	5	6	7	8	9	10	11	12	>
Picón	X	X					X	X			X		
Postero				X			X	X		X			
Tizná		X						X					
Barranco				X						X		X	



4.3.6 Consultar Oferta

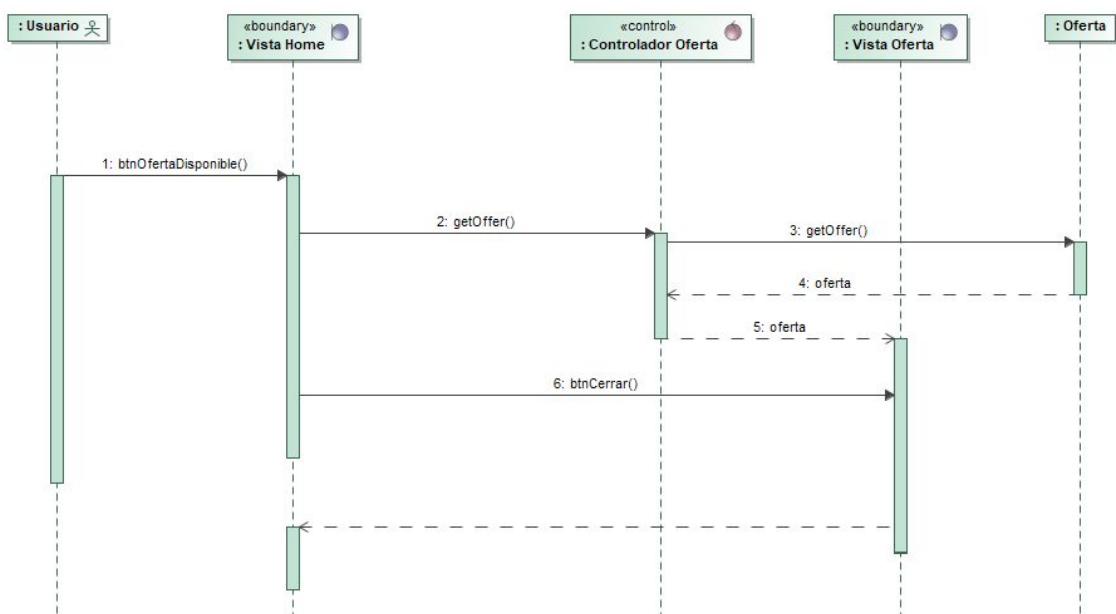
Título	Consultar Oferta
Descripción	Consultar la oferta disponible en la web
Pre-condición	No existe ninguna precondición
Post-condición	El usuario puede visualizar el título y la descripción de la oferta actual
Prioridad	Baja
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web y presiona el botón de ‘Oferta Disponible’ 2. El sistema le muestra mediante un pop-up la oferta actual	
Clases de análisis	
A. Clases de entidad	Oferta
B. Clases de control	inputdates.component.ts
C. Clases de interfaz	inputdates.component.html

Maquetas de interfaz





Diagramas de secuencia

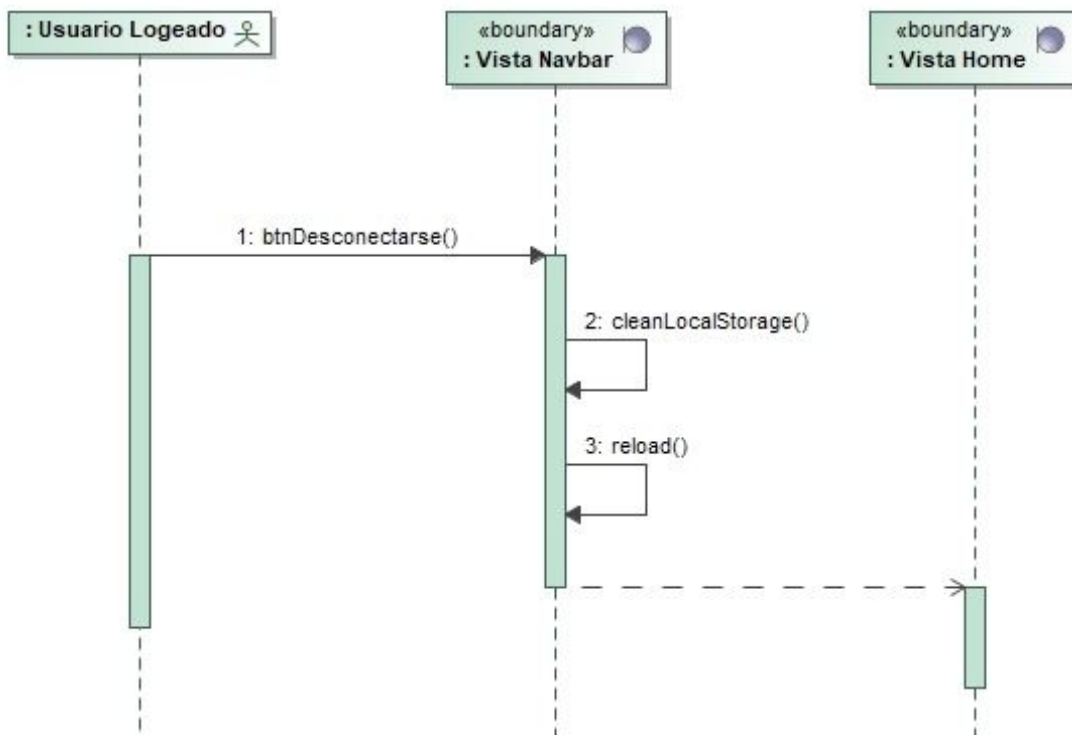


4.3.7 Log-out

Título	Log-out
Descripción	Desconectarse de la aplicación web
Pre-condición	Estar registrado y conectado en la aplicación
Post-condición	El usuario se ha desconectado y se ha eliminado el token del local storage
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Desconectarse’ de la barra de navegación 2. El usuario se ha desconectado y se ha eliminado el token del local storage	
Clases de análisis	
A. Clases de entidad	Usuario
B. Clases de control	navbar.component.ts
C. Clases de interfaz	navbar.component.html
Maquetas de interfaz	



Diagramas de secuencia

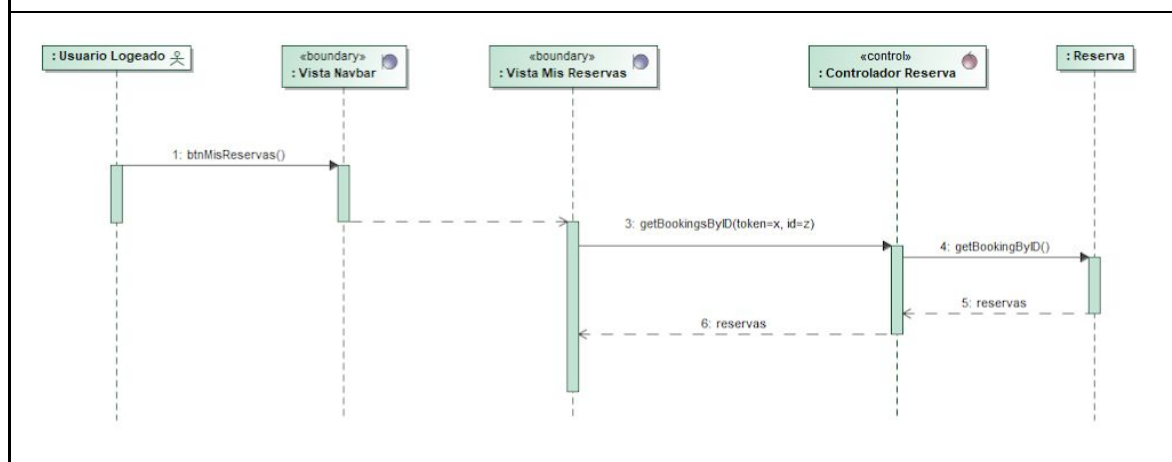


4.3.8 Consultar Reservas Usuario

Título	Consultar Reservas Usuario
Descripción	Consultar la información de las reservas que tiene un usuario concreto
Pre-condición	Estar conectado en la aplicación
Post-condición	El usuario puede visualizar las reservas que ha realizado, tanto las que ya han pasado como las próximas
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado como usuario y presiona el botón de Mis Reservas en la barra de navegación 2. El sistema le ofrece información sobre las reservas que ha realizado y la posibilidad de realizar un cuestionario de satisfacción si la reserva ya ha expirado su fecha fin	
Clases de análisis	
A. Clases de entidad	Usuario, Reserva, Cuestionario
B. Clases de control	navbar.component.ts, misreservas.component.ts
C. Clases de interfaz	navbar.component.html, misreservas.component.html
Maquetas de interfaz	

Reservar Ahora	Apartamentos	Mis Reservas
<div style="border: 1px solid black; background-color: #007bff; color: white; padding: 10px; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> Apartamentos: Picón Estado: Reserva Confirmada ✓ </div> <p>Fecha: 08/12/19 hasta 12/12/19</p> </div>		
<div style="border: 1px solid black; background-color: #007bff; color: white; padding: 10px; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> Apartamentos: Picón Estado: Pendiente de Pago ✗ </div> <p>Fecha: 21/12/19 hasta 28/12/19</p> <div style="text-align: right;"> Ver detalles </div> </div>		
<div style="border: 1px solid black; background-color: #007bff; color: white; padding: 10px;"> <div style="display: flex; justify-content: space-between;"> Apartamentos: Picón Rellenar Cuestionario Satisfacción </div> <p>Fecha: 08/12/19 hasta 12/12/19</p> </div>		

Diagramas de secuencia



4.3.9 Rellenar Cuestionario de Satisfacción

Título	Rellenar Cuestionario de Satisfacción
Descripción	Un usuario rellena el cuestionario de satisfacción de una reserva y se guarda en la base de datos

Pre-condición	Estar registrado y conectado en la aplicación, además de tener alguna reserva que no haya sido cancelada, su fecha final haya expirado y no se haya completado el cuestionario de satisfacción
Post-condición	Se ha guardado el cuestionario de satisfacción de una reserva concreta en la base de datos, que estará disponible para visualizar en el panel de administrador
Prioridad	Baja
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario accede a la web conectado y presiona el botón ‘Mis Reservas’ de la barra de navegación 2. El sistema le ofrece información sobre sus reservas y presiona el botón ‘Completar Cuestionario de Satisfacción’ en una de ellas 3. El usuario introduce los datos y presiona en el botón Enviar 4. Se ha guardado el cuestionario de satisfacción en la base de datos y el usuario vuelve a la pantalla de sus reservas 	
Escenario alternativo	
<ol style="list-style-type: none"> 1. El usuario accede a la web conectado y presiona el botón ‘Mis Reservas’ de la barra de navegación 2. El sistema le ofrece información sobre sus reservas y presiona el botón ‘Completar Cuestionario de Satisfacción’ en una de ellas 3. El usuario introduce los datos y presiona en el botón Enviar 4. No se ha introducido una puntuación a la limpieza y el sistema le avisa de que es un campo obligatorio 	
Clases de análisis	
A. Clases de entidad	Usuario, Cuestionario, Reserva
B. Clases de control	navbar.component.ts, misreservas.component.ts, cuestionario.component.ts
C. Clases de interfaz	navbar.component.html, misreservas.component.html, cuestionario.component.html
Maquetas de interfaz	

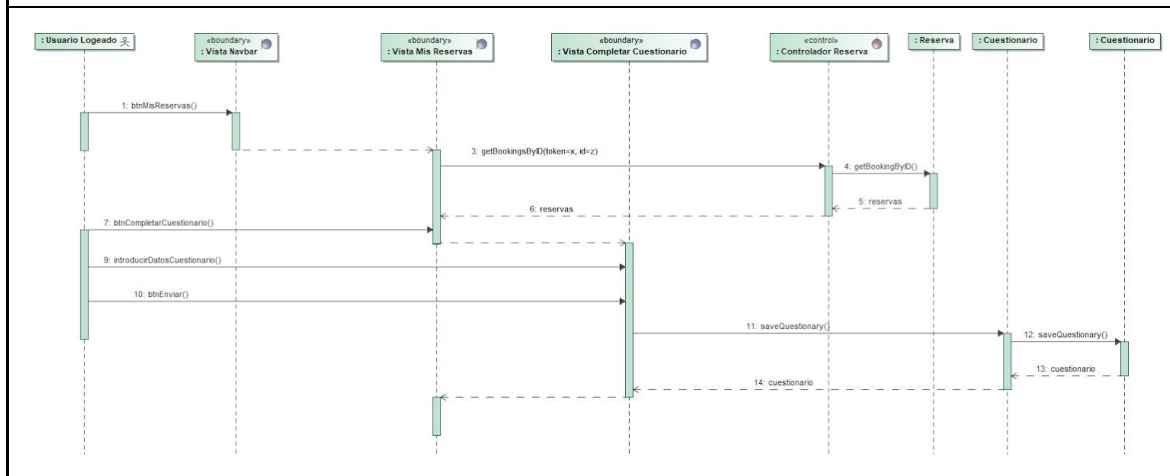
Reservar Ahora	Apartamentos	Mis Reservas
<div> <div>Apartamentos: Picón</div> <div>Estado: Reserva Confirmada </div> <div>Fecha: 08/12/19 hasta 12/12/19</div> </div>		
<div> <div>Apartamentos: Picón</div> <div>Estado: Pendiente de Pago </div> <div>Fecha: 21/12/19 hasta 28/12/19</div> <div>Ver detalles</div> </div>		
<div> <div>Apartamentos: Picón</div> <div>Fecha: 08/12/19 hasta 12/12/19</div> <div>Rellenar Cuestionario Satisfacción</div> </div>		

Reservar Ahora	Apartamentos	Mis Reservas
<div> <div>Apartamentos: Picón</div> <div>Estado: Reserva Confirmada </div> <div>Fecha: 08/12/19 hasta 12/12/19</div> </div>		
<div> <div>Apartamentos: Picón</div> <div>Fecha: 21/12/19 hasta 28/12/19</div> <div>Ver detalles</div> </div>		
<div> <div>Apartamentos: Picón</div> <div>Fecha: 08/12/19 hasta 12/12/19</div> <div>Rellenar Cuestionario Satisfacción</div> </div>		

Reservar Ahora	Apartamentos	Mis Reservas
<div> <div>Apartamentos: Picón</div> <div>Estado: Reserva Confirmada ✓</div> <div>Fecha: 08/12/19 hasta 12/12/19</div> </div>		
<div> <div>Apartamentos: Picón</div> <div>Fecha: 21/12/19 hasta 25/12/19</div> <div> <div>Limpieza</div> <div>5</div> <div>></div> </div> </div>		
<div> <div>Apartamentos: Picón</div> <div>Fecha: 08/12/19 hasta 12/12/19</div> <div>Rellenar Cuestionario Satisfacción</div> </div>		

Reservar Ahora	Apartamentos	Mis Reservas
<div> <div>Apartamentos: Picón</div> <div>Estado: Reserva Confirmada ✓</div> <div>Fecha: 08/12/19 hasta 12/12/19</div> </div>		
<div> <div>Apartamentos: Picón</div> <div>Fecha: 21/12/19 hasta 25/12/19</div> <div> <div>¿Alguna observación que quieras comentarnos.?</div> <div></div> <div>Terminar</div> </div> </div>		
<div> <div>Apartamentos: Picón</div> <div>Fecha: 08/12/19 hasta 12/12/19</div> <div>Rellenar Cuestionario Satisfacción</div> </div>		

Diagramas de secuencia



4.3.10 Filtrar Reservas

Título	Filtrar Reservas
Descripción	Consultar todas las reservas en el panel de administrador y filtrarlas por DNI de cliente
Pre-condición	Estar registrado, conectado y tener el rol de administrador
Post-condición	El usuario puede ver las reservas cuyo cliente tenga un DNI concreto
Prioridad	Baja
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020

Escenario principal

1. El usuario administrador accede a la web conectado y presiona el botón de Panel de Administrador, lo que le redirige a visualizar todas las reservas
2. El usuario introduce un DNI concreto y presiona en el botón buscar
3. El sistema le muestra todas las reservas cuyo cliente tiene el DNI introducido

Escenario alternativo

1. El usuario administrador accede a la web conectado y presiona el botón de Panel de Administrador, lo que le redirige a visualizar todas las reservas
2. El usuario introduce un DNI concreto y presiona en el botón buscar
3. No existe ninguna reserva con esas características y el sistema le muestra una notificación

Clases de análisis

A. Clases de entidad	Usuario, Reserva, Apartamento
B. Clases de control	navbar.component.ts, panel-admin.component.ts, reservas-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html, reservas-admin.component.html

Maquetas de interfaz

Maqueta de interfaz de usuario para 'Alojamientos Rurales'. La interfaz muestra una barra de navegación superior con los siguientes elementos:

- Barra de navegación: Tarifas & Disponibilidad, Reservas, Apartamentos, Cuestionarios, Oferta y Anticipos.
- Barra de búsqueda: http://ruralalojamientos.com/administrador.
- Botones de acción: Añadir Reserva, Añadir Cliente.

El contenido principal se divide en dos secciones de reserva:

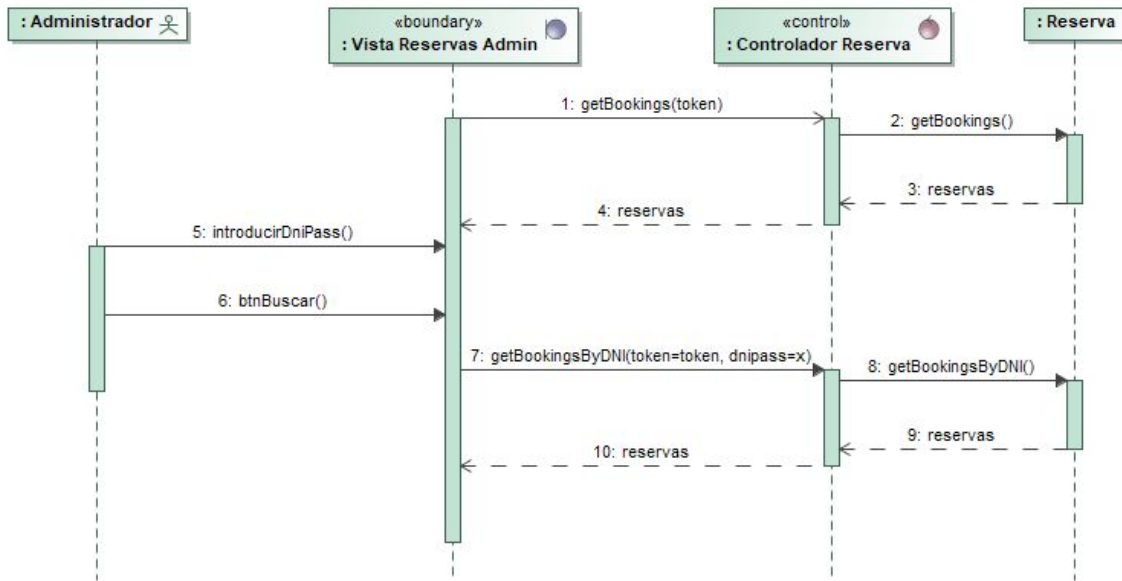
Reserva 1:

- Apartamentos: Picón
- Fecha: 08/12/19 hasta 12/12/19
- Estado: Pendiente de Pago
- Valoración: Seleccionar
- Adultos: 2
- Menores: 1
- Nombre: Alberto Sánchez
- DNI: 44568874L
- Telefono: 666555666
- E-Mail: albertorg@uma.es
- Nacimiento: 07/01/1965
- Botón: Guardar Cambios

Reserva 2:

- Apartamentos: Tizná
- Fecha: 08/12/19 hasta 12/12/19
- Estado: Pendiente de Pago
- Valoración: Seleccionar
- Adultos: 2
- Menores: 1
- Nombre: María Figueroa
- DNI: 36674892M
- Telefono: 603455678
- E-Mail: mariaf@yahoo.com

Diagramas de secuencia



4.3.11 Validar Reserva

Título	Validar Reserva
Descripción	Confirmar una reserva mediante el panel de administrador
Pre-condición	Estar registrado, conectado y tener el rol de administrador
Post-condición	La reserva se ha modificado en la base de datos y su campo 'Estado' ha cambiado a 'Confirmada'
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario administrador accede a la web conectado y presiona el botón de Panel de Administrador, lo que le redirige a visualizar todas las reservas 2. El usuario abre el abanico de 'Estado' de una de las reservas y selecciona la opción 'Confirmada' y pulsa el botón 'Guardar Cambios' 3. La reserva se ha modificado correctamente y su nuevo estado es 'Confirmada'	

Clases de análisis

A. Clases de entidad	Usuario, Reserva, Apartamento
B. Clases de control	navbar.component.ts, panel-admin.component.ts, reservas-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html, reservas-admin.component.html

Maquetas de interfaz

Alojamientos Rurales

http://ruralalojamientos.com/administrador

Tarifas & Disponibilidad Reservas Apartamentos Cuestionarios Oferta y Anticipos

DNI/PASS: 44678874L Filtrar

Añadir Reserva

Añadir Cliente

Apartamentos: Picón Estado: Pendiente de Pago

Fecha: 08/12/19 hasta 12/12/19 Valoración: Seleccionar

Nombre: Alberto Sánchez
DNI: 44568874L
Telefono: 666555666
E-Mail: albertorg@uma.es
Nacimiento: 07/01/1965

Adultos: 2
Menores: 1

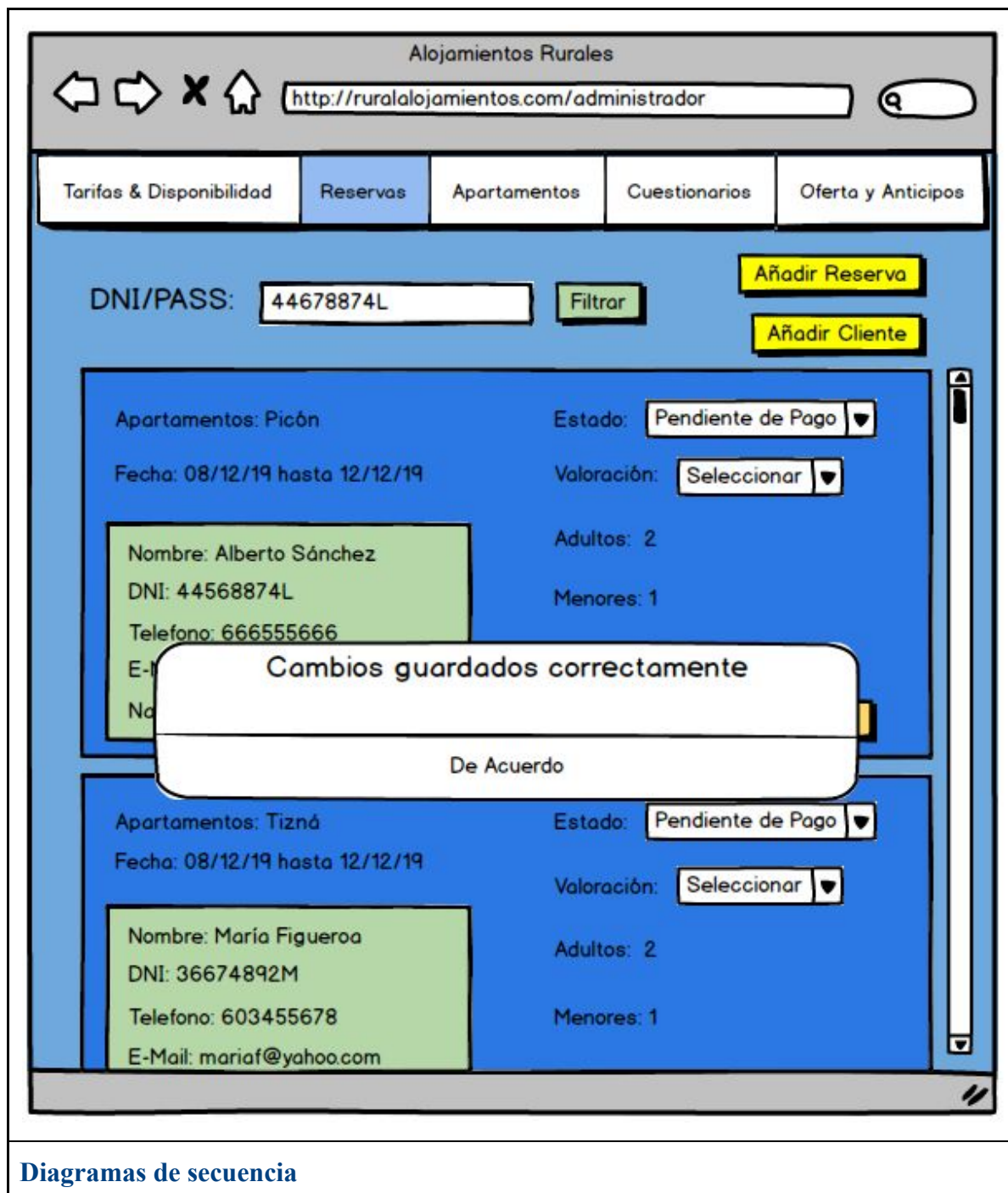
Guardar Cambios

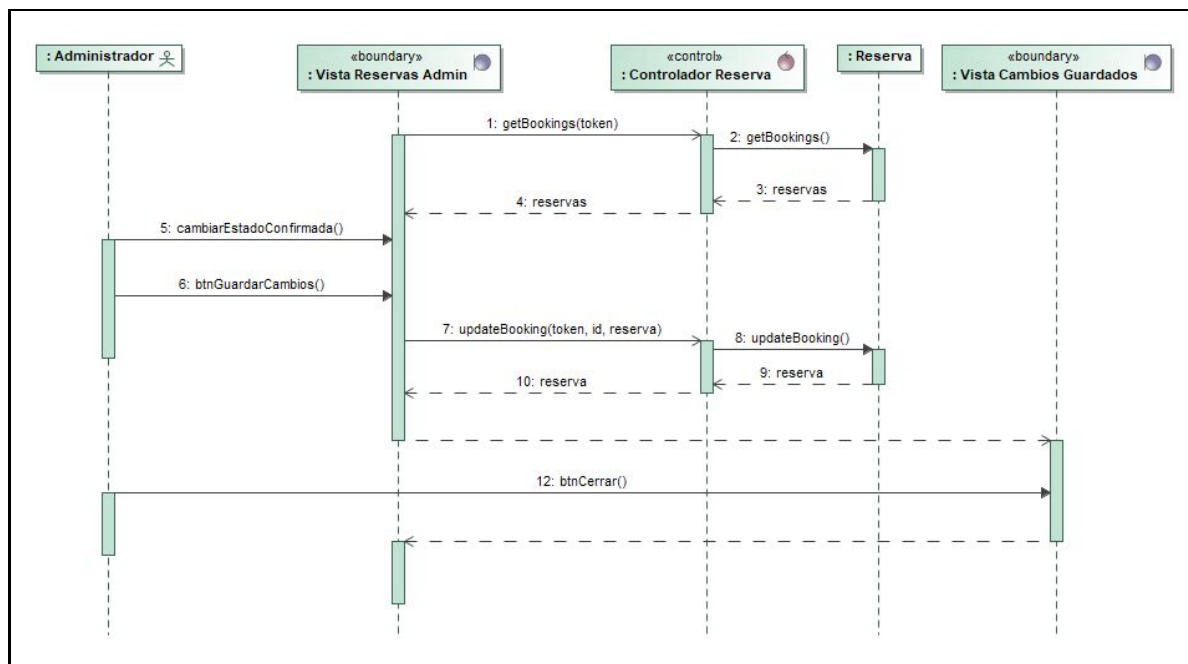
Apartamentos: Tizná Estado: Pendiente de Pago

Fecha: 08/12/19 hasta 12/12/19 Valoración: Seleccionar

Nombre: María Figueroa
DNI: 36674892M
Telefono: 603455678
E-Mail: mariaf@yahoo.com

Adultos: 2
Menores: 1





4.3.12 Valorar Reserva

Título	Valorar Reserva
Descripción	Valorar una reserva con un comentario mediante el panel de administrador
Pre-condición	Estar registrado, conectado y tener el rol de administrador
Post-condición	La reserva se ha modificado en la base de datos, en concreto el campo 'Valoración'
Prioridad	Baja
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario administrador accede a la web conectado y presiona el botón de 'Panel de Administrador', lo que le redirige a visualizar todas las reservas 2. El usuario escribe un comentario en el campo 'Valoración' de alguna de las reservas y presiona el botón 'Guardar Cambios' 3. La reserva se ha modificado correctamente y su campo 'Valoración' ha cambiado	
Clases de análisis	
A. Clases de entidad	Usuario, Reserva, Apartamento

B. Clases de control	navbar.component.ts, panel-admin.component.ts, reservas-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html, reservas-admin.component.html

Maquetas de interfaz

Alojamientos Rurales

http://ruralalojamientos.com/administrador

Tarifas & Disponibilidad Reservas Apartamentos Cuestionarios Oferta y Anticipos

DNI/PASS: 44678874L Filtrar Añadir Reserva Añadir Cliente

Apartamentos: Picón Estado: Pendiente de Pago
Fecha: 08/12/19 hasta 12/12/19 Valoración: Seleccionar
Nombre: Alberto Sánchez
DNI: 44568874L
Telefono: 666555666
E-Mail: albertorg@uma.es
Nacimiento: 07/01/1965
Adultos: 2
Menores: 1
Guardar Cambios

Apartamentos: Tizná Estado: Pendiente de Pago
Fecha: 08/12/19 hasta 12/12/19 Valoración: Seleccionar
Nombre: María Figueroa
DNI: 36674892M
Telefono: 603455678
E-Mail: mariaf@yahoo.com
Adultos: 2
Menores: 1

Alojamientos Rurales

http://ruralalojamientos.com/administrador

Tarifas & Disponibilidad

Reservas

Apartamentos

Cuestionarios

Oferta y Anticipos

DNI/PASS: 44678874L

Filtrar

Añadir Reserva

Añadir Cliente

Apartamentos: Picón

Estado: Pendiente de Pago

Fecha: 08/12/19 hasta 12/12/19

Valoración: Seleccionar

Nombre: Alberto Sánchez

DNI: 44568874L

Telefono: 666555666

E-Mail:

Na

Adultos: 2

Menores: 1

Cambios guardados correctamente

De Acuerdo

Apartamentos: Tizná

Estado: Pendiente de Pago

Fecha: 08/12/19 hasta 12/12/19

Valoración: Seleccionar

Nombre: María Figueroa

DNI: 36674892M

Telefono: 603455678

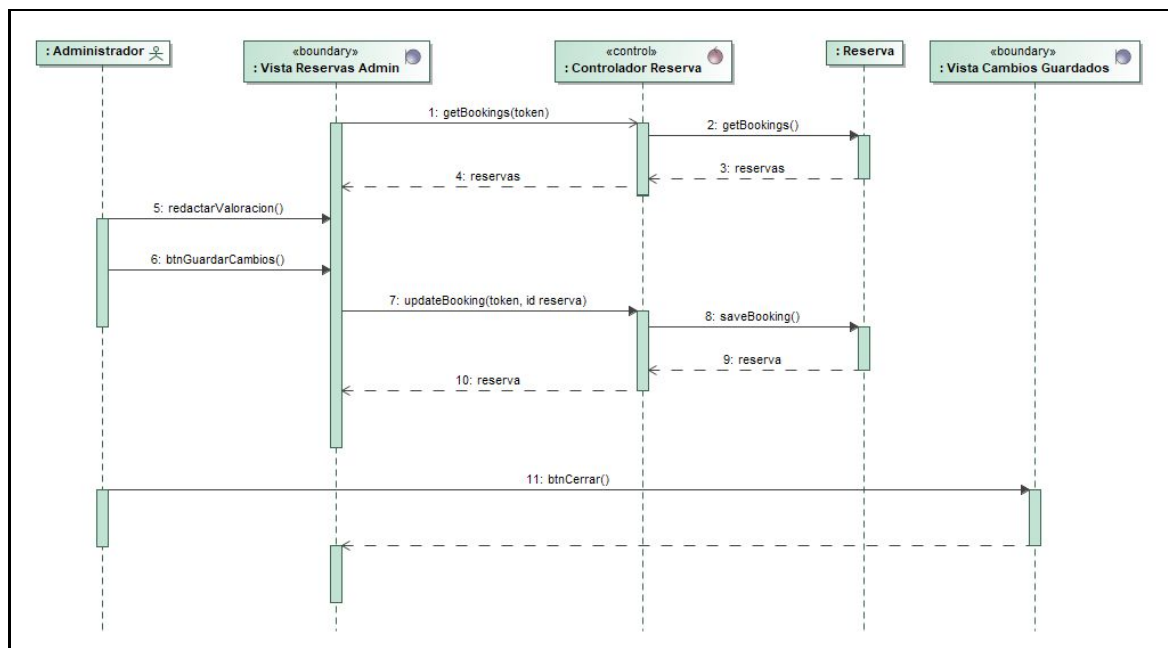
E-Mail: mariaf@yahoo.com

Adultos: 2

Menores: 1

Diagramas de secuencia

68



4.3.13 Consultar Cuestionarios de Satisfacción

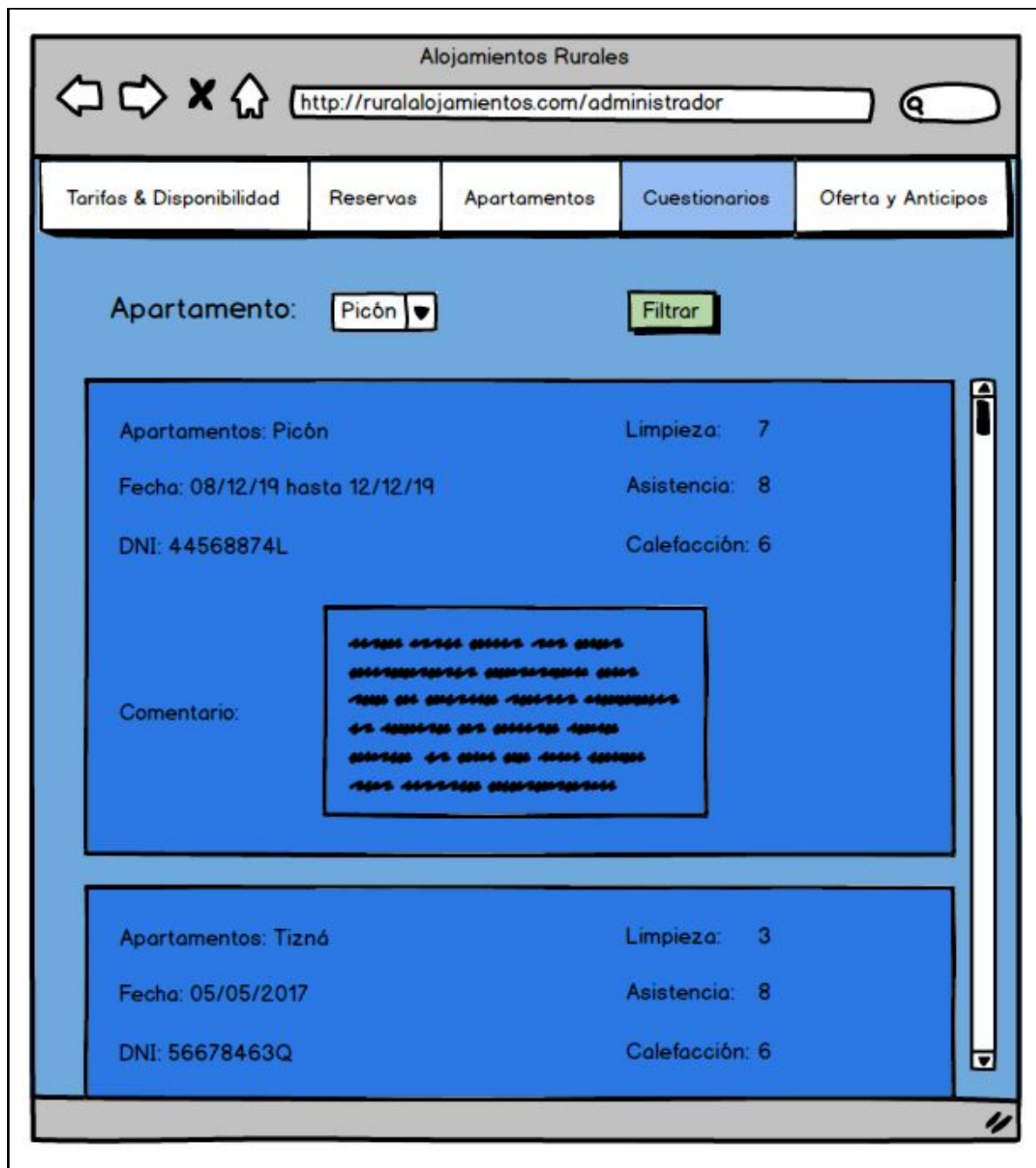
Título	Consultar Cuestionarios de Satisfacción
Descripción	Un usuario administrador consulta los cuestionarios de satisfacción que los clientes han rellenado de sus reservas
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema le muestra al usuario administrador todos los cuestionarios que se han rellenado
Prioridad	Media
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Cuestionarios’ para acceder a la página correspondiente 3. El sistema le muestra todos los cuestionarios que existen en la base de datos y su información	
Escenario alternativo	

1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación
2. El usuario presiona el botón ‘Cuestionarios’ para acceder a la página correspondiente
3. No existe ningún cuestionario y el sistema muestra una notificación

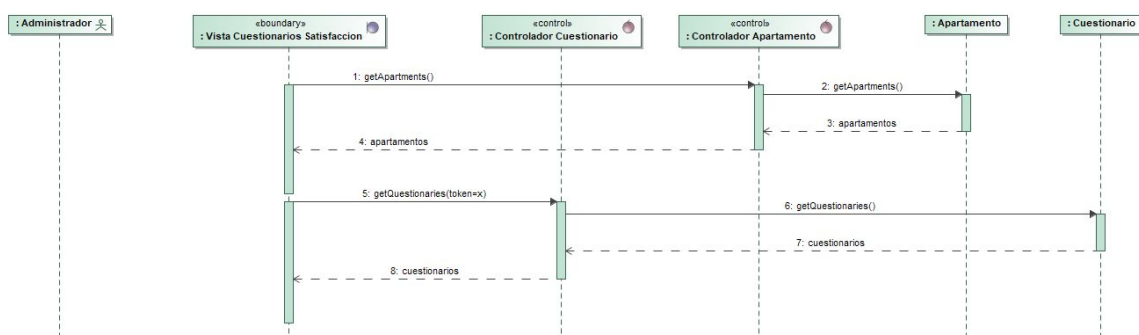
Clases de análisis

A. Clases de entidad	Usuario, Cuestionario, Reserva, Apartamento
B. Clases de control	navbar.component.ts, panel-admin.component.ts, cuestionarios-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html cuestionarios-admin.component.html

Maquetas de interfaz



Diagramas de secuencia



4.3.14 Filtrar Cuestionarios de Satisfacción

Título	Filtrar Cuestionarios de Satisfacción
Descripción	Un usuario administrador consulta los cuestionarios de satisfacción que los clientes han rellenado de sus reservas y los filtra por apartamento
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema le muestra al usuario administrador todos los cuestionarios que se han rellenado filtrados por apartamento
Prioridad	Baja
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Cuestionarios’ para acceder a la página correspondiente 3. El usuario selecciona un apartamento para filtrar y presiona en el botón ‘Buscar’ 3. El sistema le muestra todos los cuestionarios de las reservas que existen en la base de datos y su información en cuyos apartamentos aparezca el que ha seleccionado	
Escenario alternativo	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Cuestionarios’ para acceder a la página correspondiente 3. El usuario selecciona un apartamento para filtrar y presiona en el botón ‘Buscar’ 3. No existe ningún cuestionario cuya reserva contenga el apartamento que se ha seleccionado para filtrar y se muestra una notificación	
Clases de análisis	
A. Clases de entidad	Usuario, Cuestionario, Reserva, Apartamento
B. Clases de control	navbar.component.ts, panel-admin.component.ts, cuestionarios-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html

Maquetas de interfaz

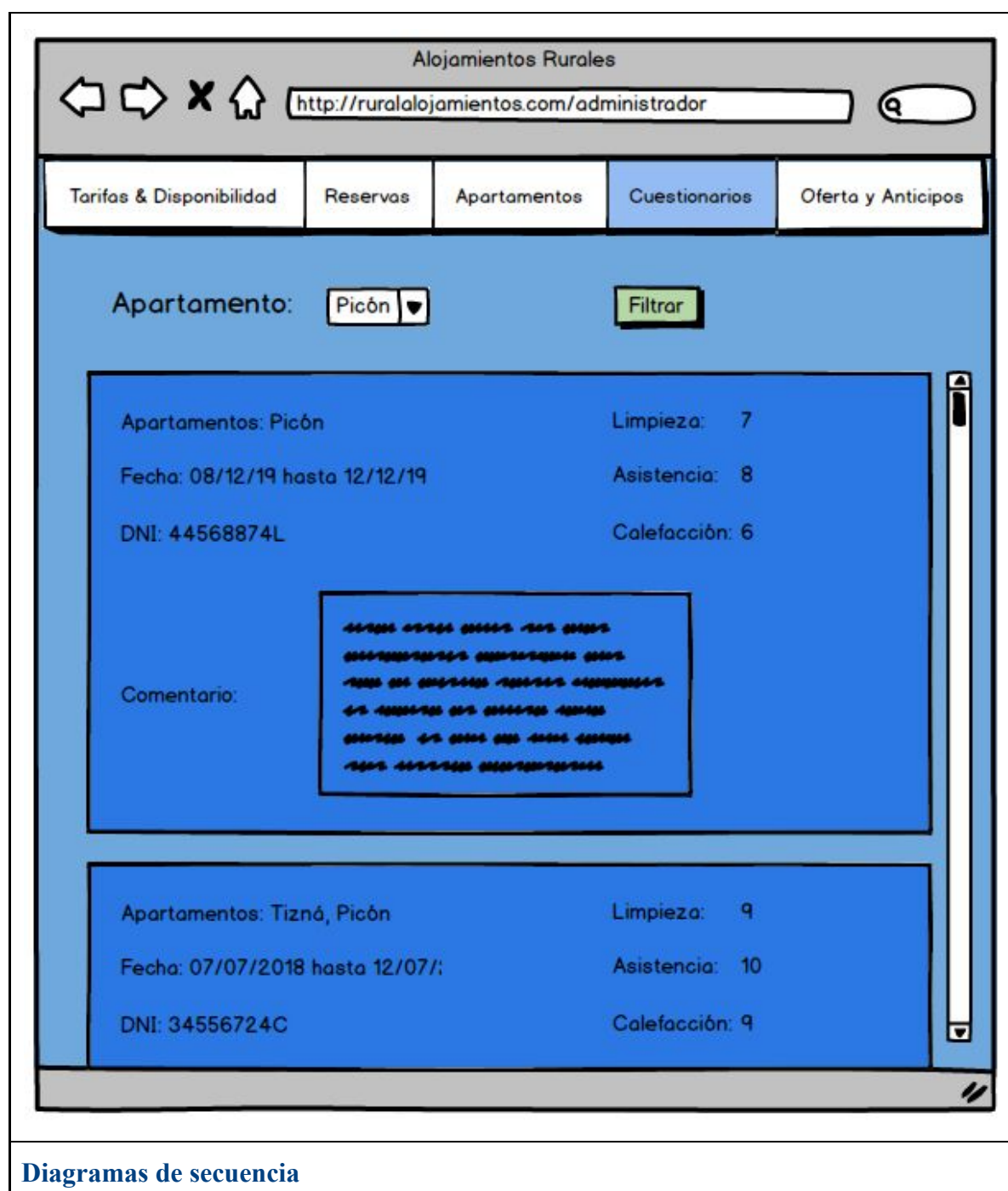
The mockup shows a web browser window titled 'Alojamientos Rurales' with the URL 'http://ruralalojamientos.com/administrador'. The navigation bar includes tabs for 'Tarifas & Disponibilidad', 'Reservas', 'Apartamentos', 'Cuestionarios' (selected), and 'Oferta y Anticipos'. The main content area is titled 'Apartamento:' and features a dropdown menu set to 'Picón' and a 'Filtrar' button. Below this, there are two main sections for data entry:

Section 1 (Picón):

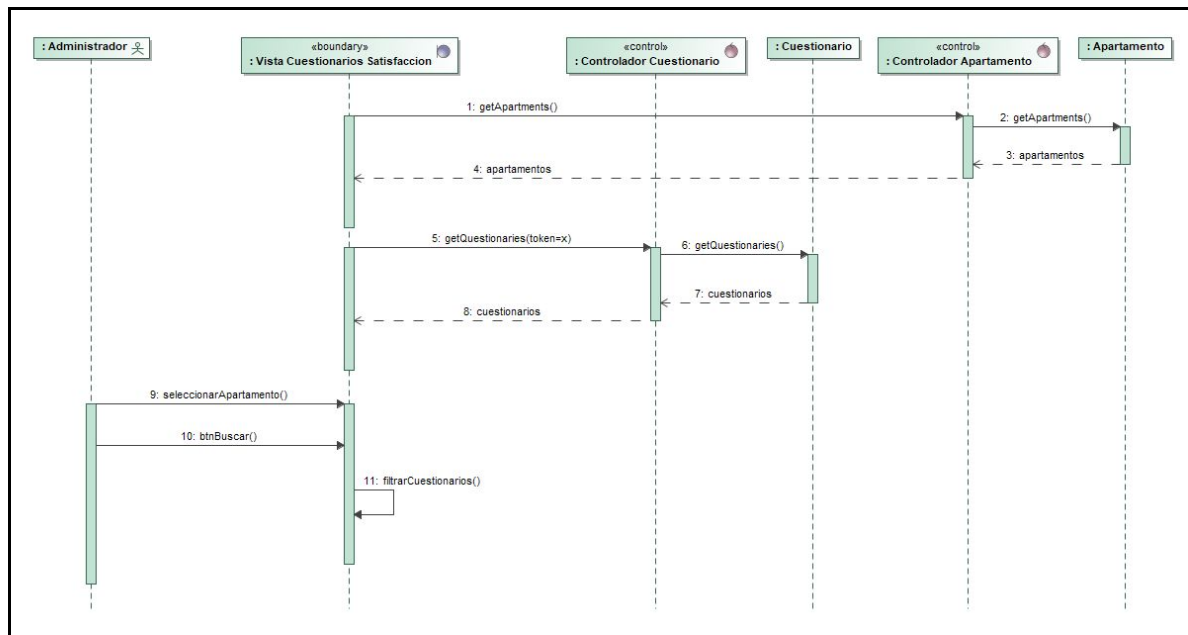
- Apartamentos: Picón
- Fecha: 08/12/19 hasta 12/12/19
- DNI: 44568874L
- Limpieza: 7
- Asistencia: 8
- Calefacción: 6
- Comentario: (Text area with placeholder text)

Section 2 (Tizná):

- Apartamentos: Tizná
- Fecha: 05/05/2017
- DNI: 56678463Q
- Limpieza: 3
- Asistencia: 8
- Calefacción: 6



Diagramas de secuencia



4.3.15 Actualizar Oferta

Título	Actualizar Oferta
Descripción	Un usuario administrador modifica la oferta actual
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	La oferta ha sido modificada en la base de datos
Prioridad	Media
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Oferta & Anticipos’ para acceder a la página correspondiente 3. El usuario introduce los nuevos datos de la oferta y pulsa en el botón ‘Actualizar Oferta’ 4. La oferta ha sido modificada en la base de datos	
Clases de análisis	
A. Clases de entidad	Usuario, Oferta

B. Clases de control	navbar.component.ts, panel-admin.component.ts, oferta-anticipos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html oferta-anticipos-admin.component.html

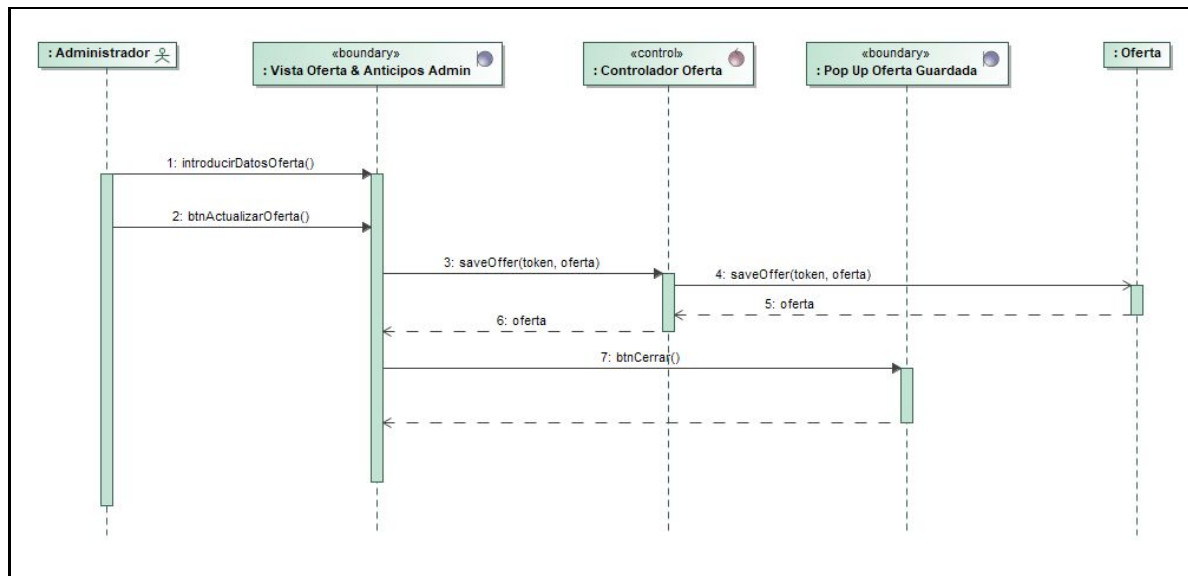
Maquetas de interfaz

The mockup shows a web browser window titled 'Alojamientos Rurales' with the URL 'http://ruralalojamientos.com/administrador'. The navigation bar includes tabs for 'Tarifas & Disponibilidad', 'Reservas', 'Apartamentos', 'Cuestionarios', and 'Oferta y Anticipos' (which is active). The main content area has a 'Titulo:' label followed by a text input field and a 'Guardar Cambios' button. Below this is a 'Texto:' label followed by a larger text area. A radio button option 'No hay ninguna oferta actualmente' is present. A table titled 'Fechas con Anticipo' contains the following data:

Fechas de Inicio	Fecha de Fin	Eliminar
21/12/2019	09/01/2020	X

At the bottom, there is a form with 'Desde' (21/12/2019) and 'Hasta' (09/01/2020) date pickers, and an 'Añadir' button.

Diagramas de secuencia

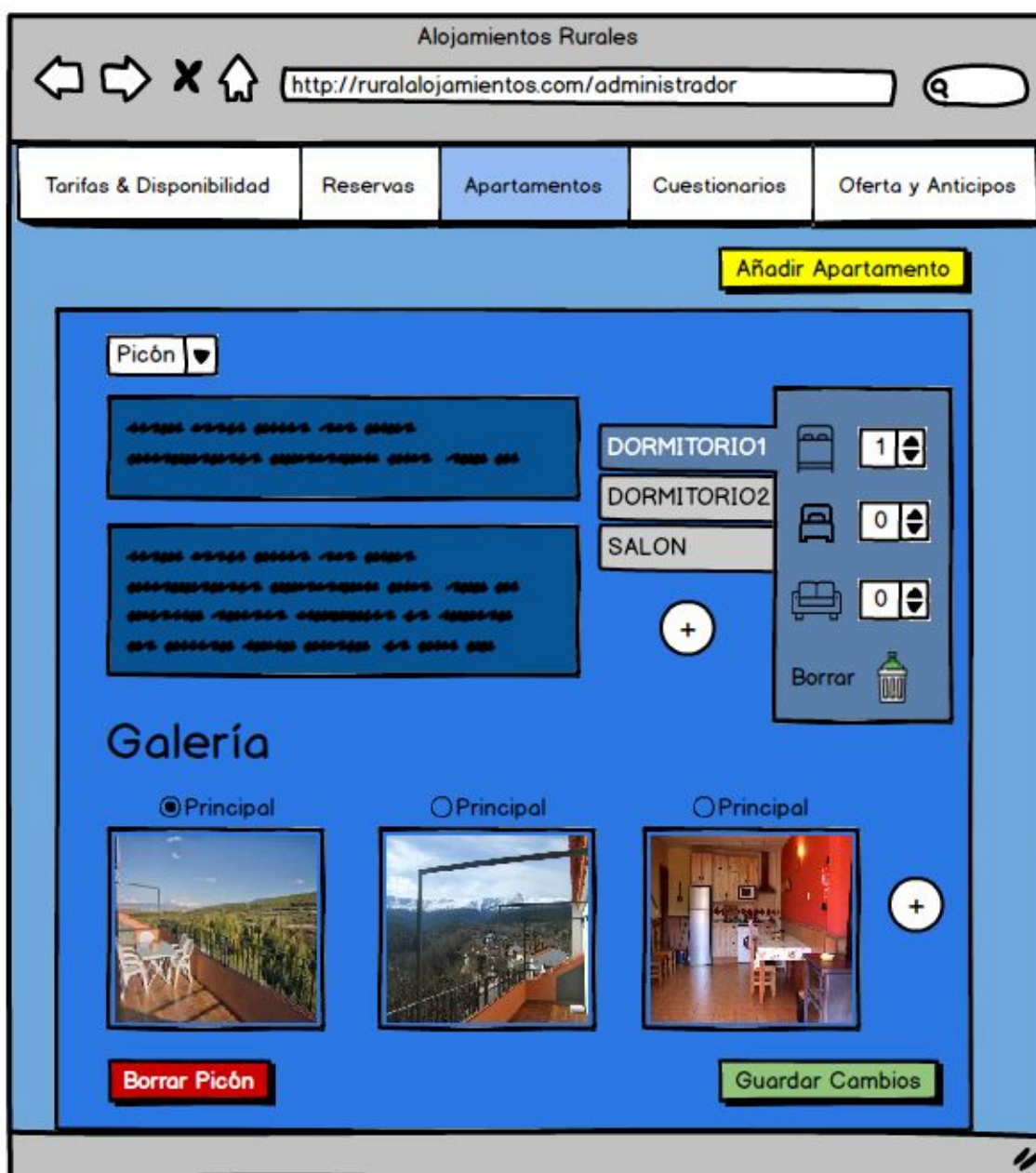


4.3.16 Consultar Apartamento

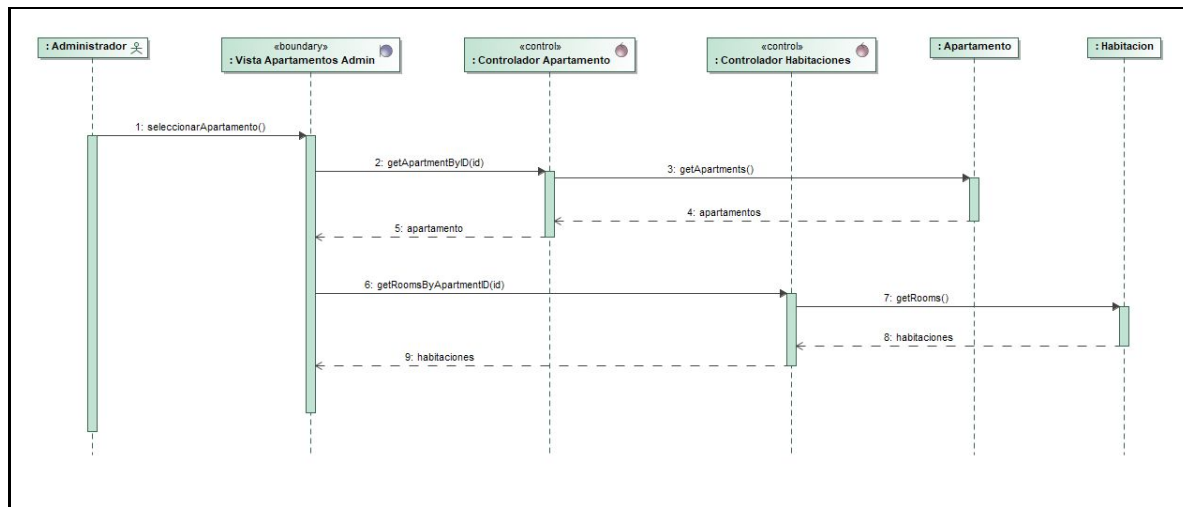
Título	Consultar Apartamento
Descripción	Un usuario administrador visualiza un apartamento en su panel de administrador
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ofrece al usuario la información acerca del apartamento
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Apartamentos’ para acceder a la página correspondiente 3. El sistema le ofrece al usuario un scroll para seleccionar el apartamento a visualizar y carga el primero de la lista inicialmente	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Habitación
B. Clases de control	navbar.component.ts,

	panel-admin.component.ts, apartamentos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html apartamentos-admin.component.html

Maquetas de interfaz



Diagramas de secuencia

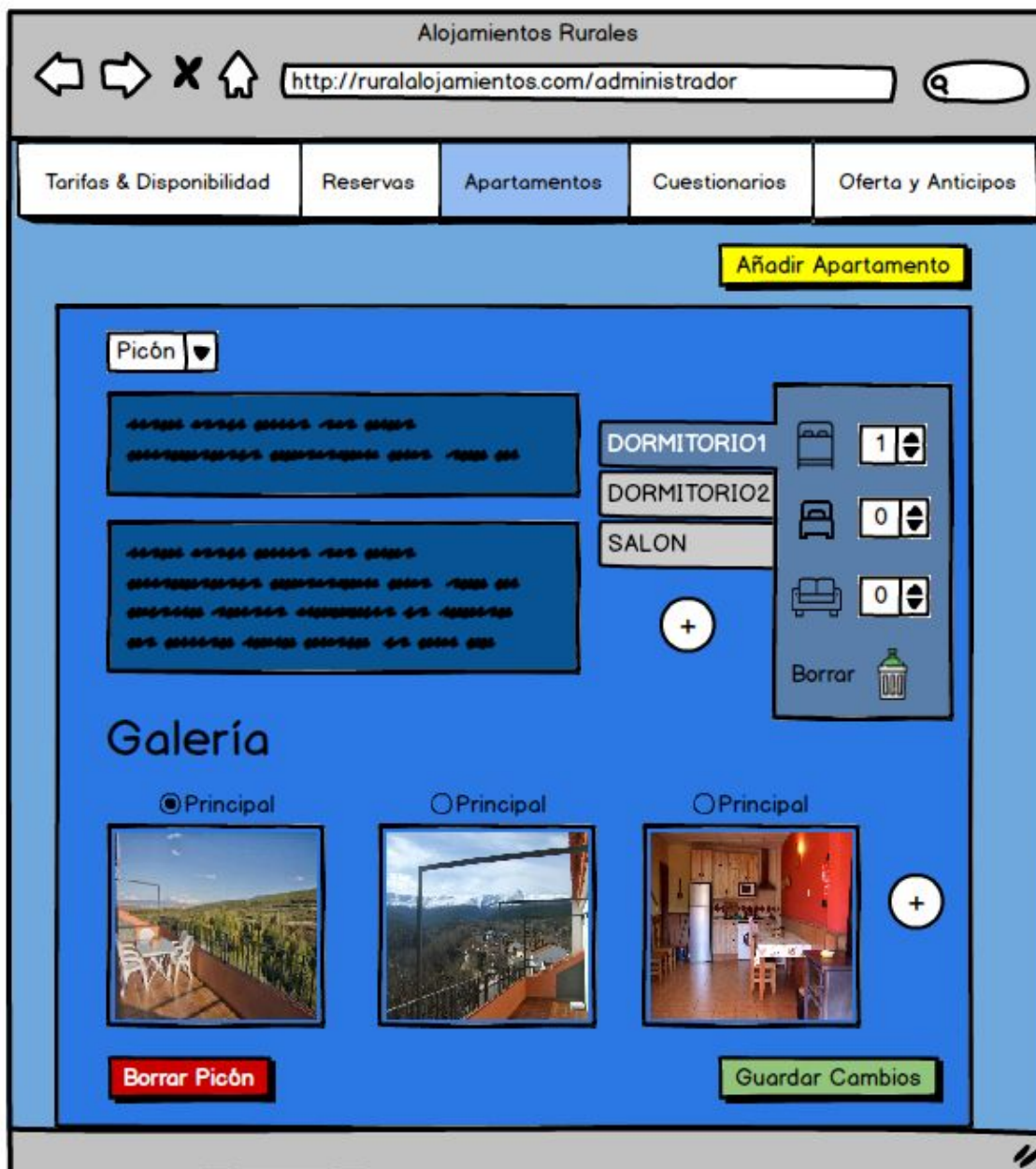


4.3.17 Modificar Apartamento

Título	Modificar Apartamento
Descripción	Un usuario administrador modifica un apartamento en su panel de administrador
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema modifica el apartamento en la base de datos
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Apartamentos’ para acceder a la página correspondiente 3. El sistema le ofrece al usuario un scroll para seleccionar el apartamento a visualizar y carga el primero de la lista inicialmente 4. El usuario modifica la descripción del apartamento y presiona el botón ‘Guardar Cambios’ 5. El apartamento ha sido modificado correctamente en la base de datos y notifica con un Pop Up	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Habitación

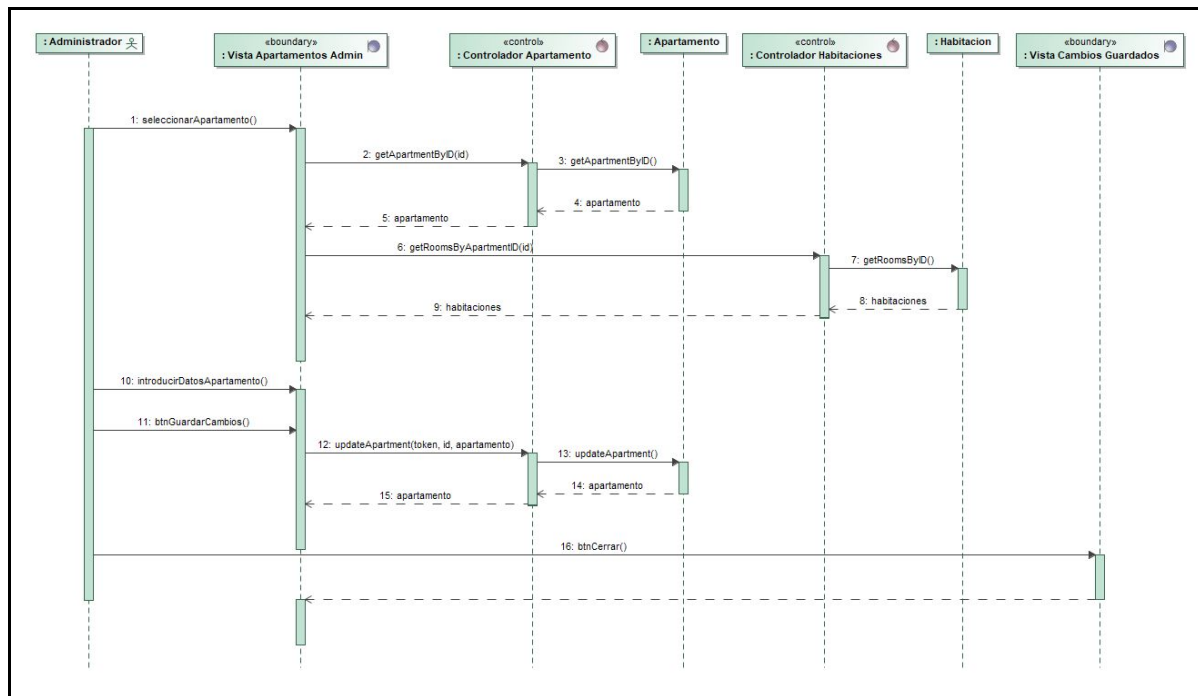
B. Clases de control	navbar.component.ts, panel-admin.component.ts, apartamentos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html apartamentos-admin.component.html

Maquetas de interfaz





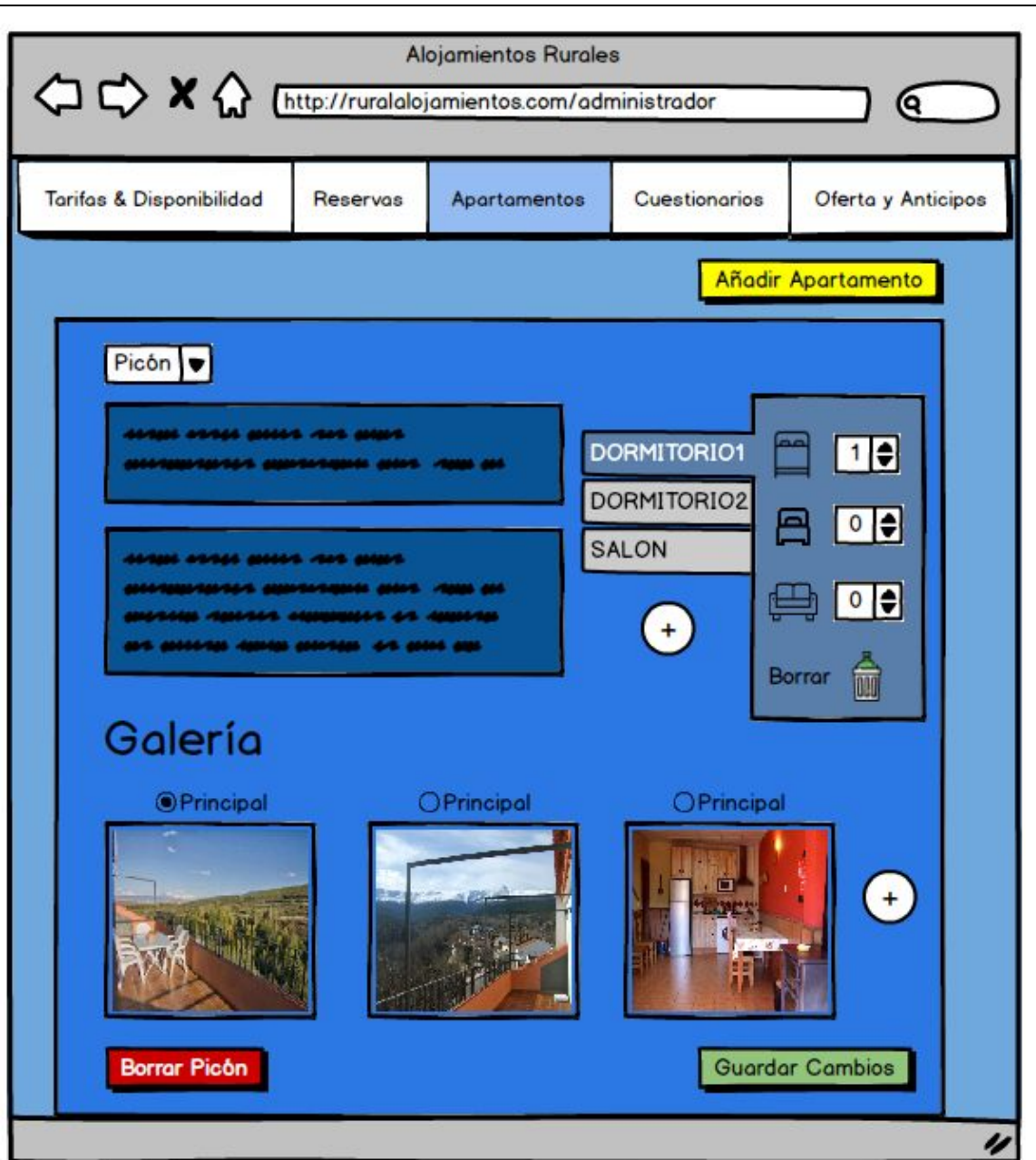
Diagramas de secuencia



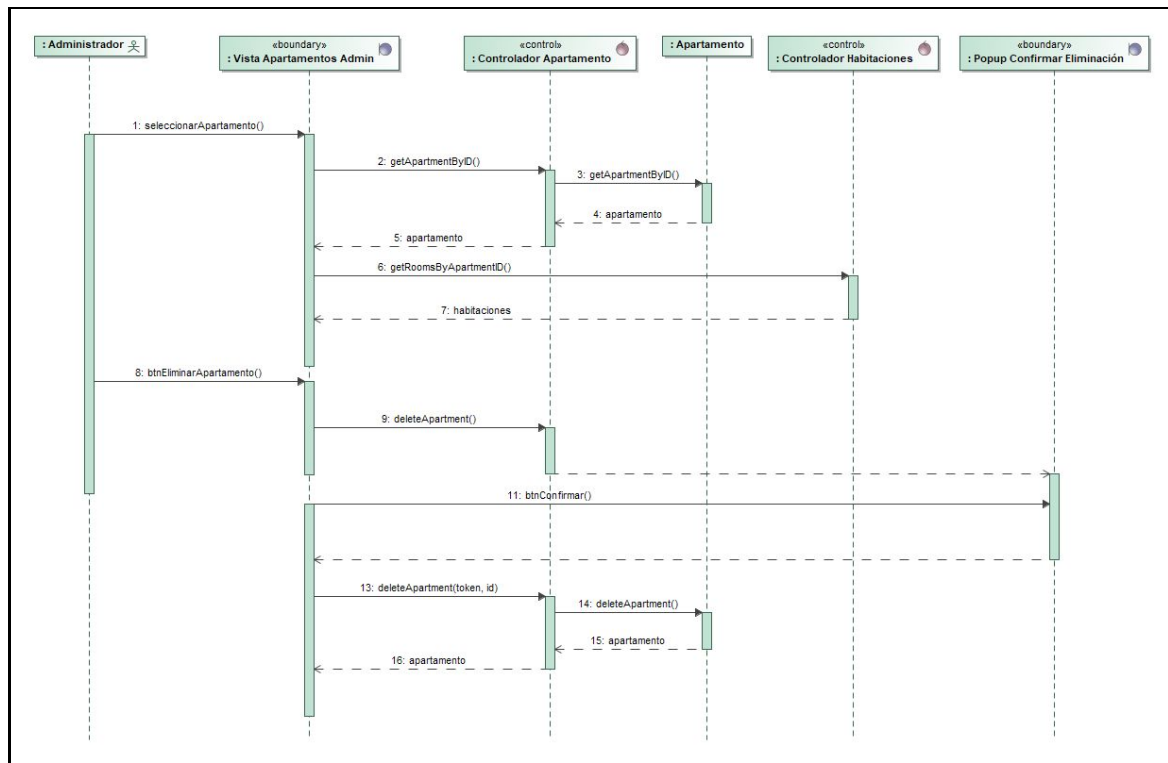
4.3.18 Eliminar Apartamento

Título	Eliminar Apartamento
Descripción	Un usuario administrador modifica un apartamento en su panel de administrador
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema elimina el apartamento en la base de datos
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Apartamentos’ para acceder a la página correspondiente 3. El sistema le ofrece al usuario un scroll para seleccionar el apartamento a visualizar y carga el primero de la lista inicialmente 4. El usuario presiona el botón ‘Eliminar Apartamento’ 5. El sistema pregunta al usuario de si realmente quiere eliminar el apartamento y le da la posibilidad de cancelar el proceso 6. El usuario confirma la eliminación del apartamento y es eliminado de la base de datos 	

Escenario alternativo	
<ol style="list-style-type: none"> 1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Apartamentos’ para acceder a la página correspondiente 3. El sistema le ofrece al usuario un scroll para seleccionar el apartamento a visualizar y carga el primero de la lista inicialmente 4. El usuario presiona el botón ‘Eliminar Apartamento’ 5. El sistema pregunta al usuario de si realmente quiere eliminar el apartamento y le da la posibilidad de cancelar el proceso 6. El usuario cancela la eliminación del apartamento y vuelve a la pantalla de edición 	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Habitación
B. Clases de control	navbar.component.ts, panel-admin.component.ts, apartamentos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html apartamentos-admin.component.html
Maquetas de interfaz	



Diagramas de secuencia



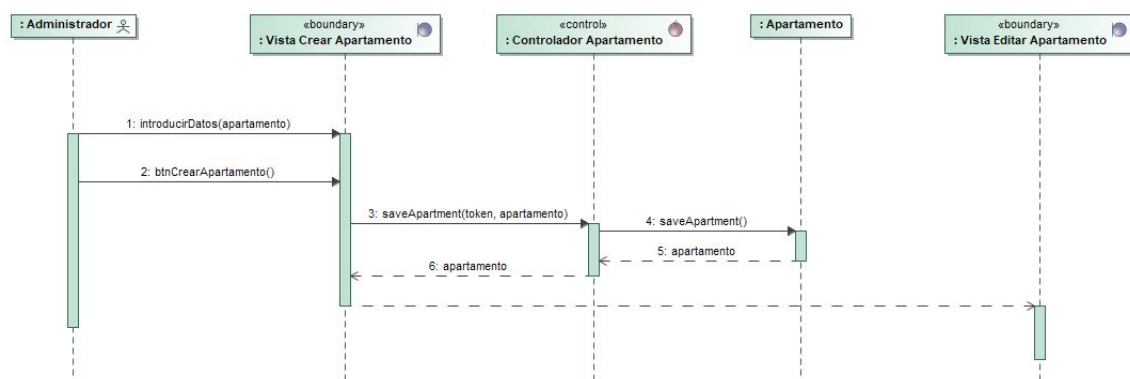
4.3.19 Crear Apartamento

Título	Crear Apartamento
Descripción	Un usuario administrador modifica un apartamento en su panel de administrador
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema crea el apartamento en la base de datos
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Apartamentos’ para acceder a la página correspondiente 3. El sistema le ofrece al usuario un scroll para seleccionar el apartamento a visualizar y carga el primero de la lista inicialmente 4. El usuario presiona el botón ‘Añadir Apartamento’ 	

<p>5. El usuario introduce los datos del nuevo apartamento y presiona el botón ‘Crear Apartamento’</p> <p>6. El apartamento ha sido creado correctamente y envía al usuario a la pantalla para editarlo</p>	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Habitación
B. Clases de control	navbar.component.ts, panel-admin.component.ts, apartamentos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html apartamentos-admin.component.html
Maquetas de interfaz	



Diagramas de secuencia

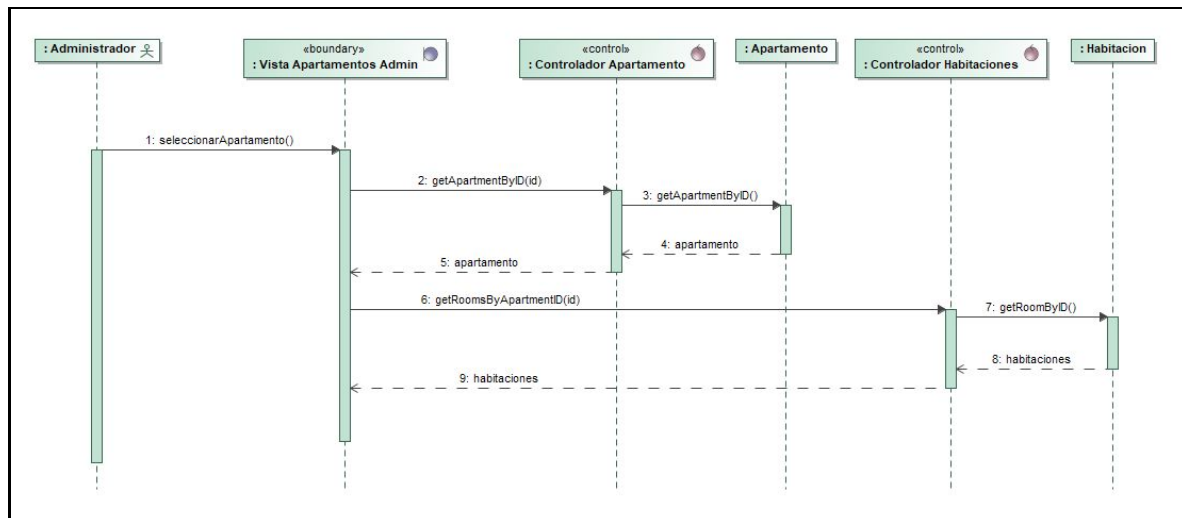


4.3.20 Consultar Habitaciones

Título	Consultar Habitación
Descripción	Un usuario administrador visualiza las habitaciones de un apartamento en su panel de administrador
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema muestra al usuario administrador la información de las habitaciones de un apartamento concreto
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Apartamentos’ para acceder a la página correspondiente 3. El sistema le ofrece al usuario un scroll para seleccionar el apartamento a visualizar y carga el primero de la lista inicialmente 4. El sistema muestra al usuario la información de las habitaciones del apartamento seleccionado	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Habitación
B. Clases de control	navbar.component.ts, panel-admin.component.ts, apartamentos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html apartamentos-admin.component.html
Maquetas de interfaz	



Diagramas de secuencia



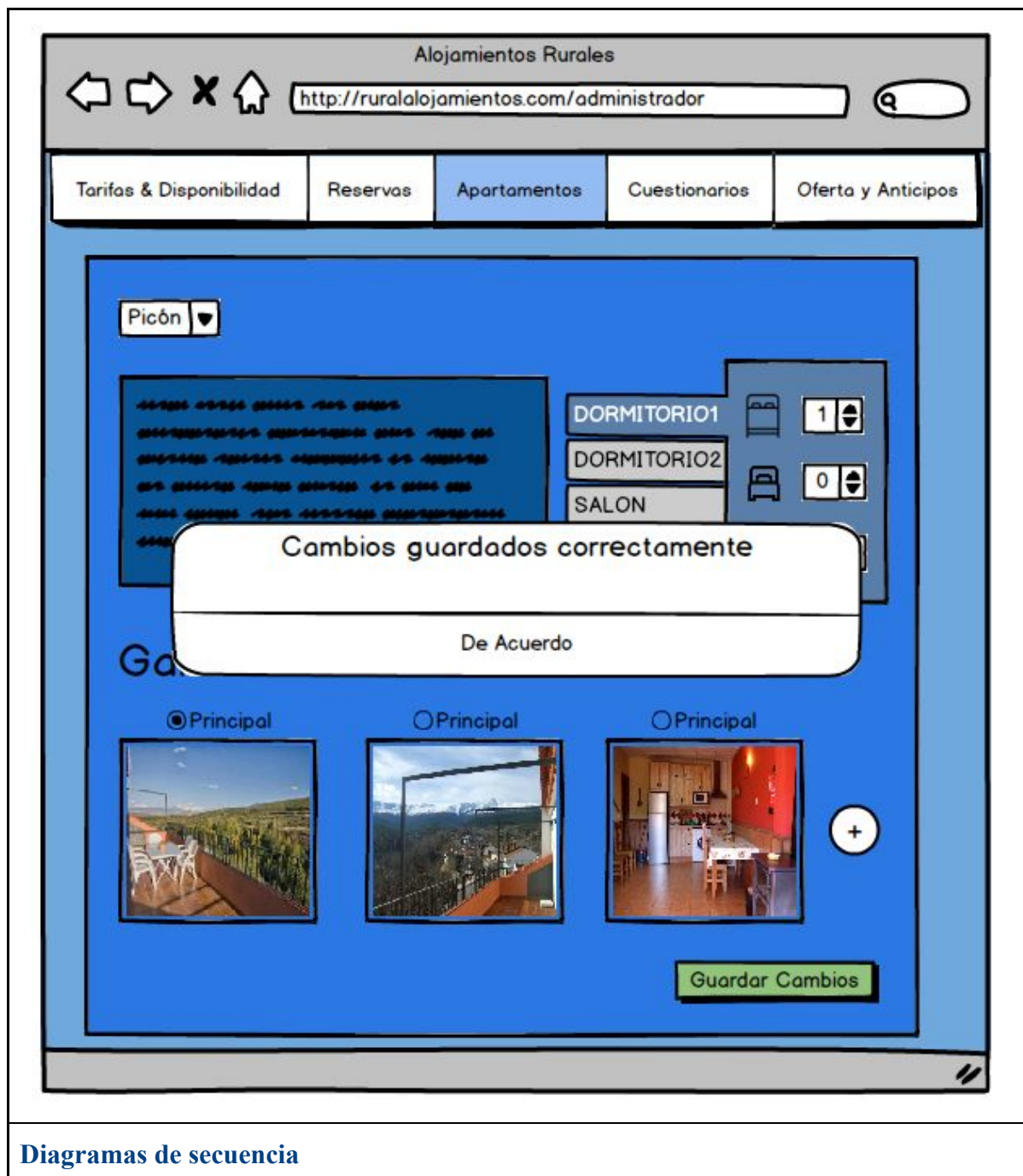
4.3.21 Modificar Habitaciones

Título	Modificar Habitaciones
Descripción	Un usuario administrador modifica las habitaciones de un apartamento en su panel de administrador
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema modifica las habitaciones del apartamento en la base de datos
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Apartamentos’ para acceder a la página correspondiente 3. El sistema le ofrece al usuario un scroll para seleccionar el apartamento a visualizar y carga el primero de la lista inicialmente 4. El usuario modifica el número de camas de matrimonio de una habitación del apartamento y presiona el botón ‘Guardar Cambios’ 5. El apartamento ha sido modificado correctamente en la base de datos y notifica con un Pop Up	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Habitación

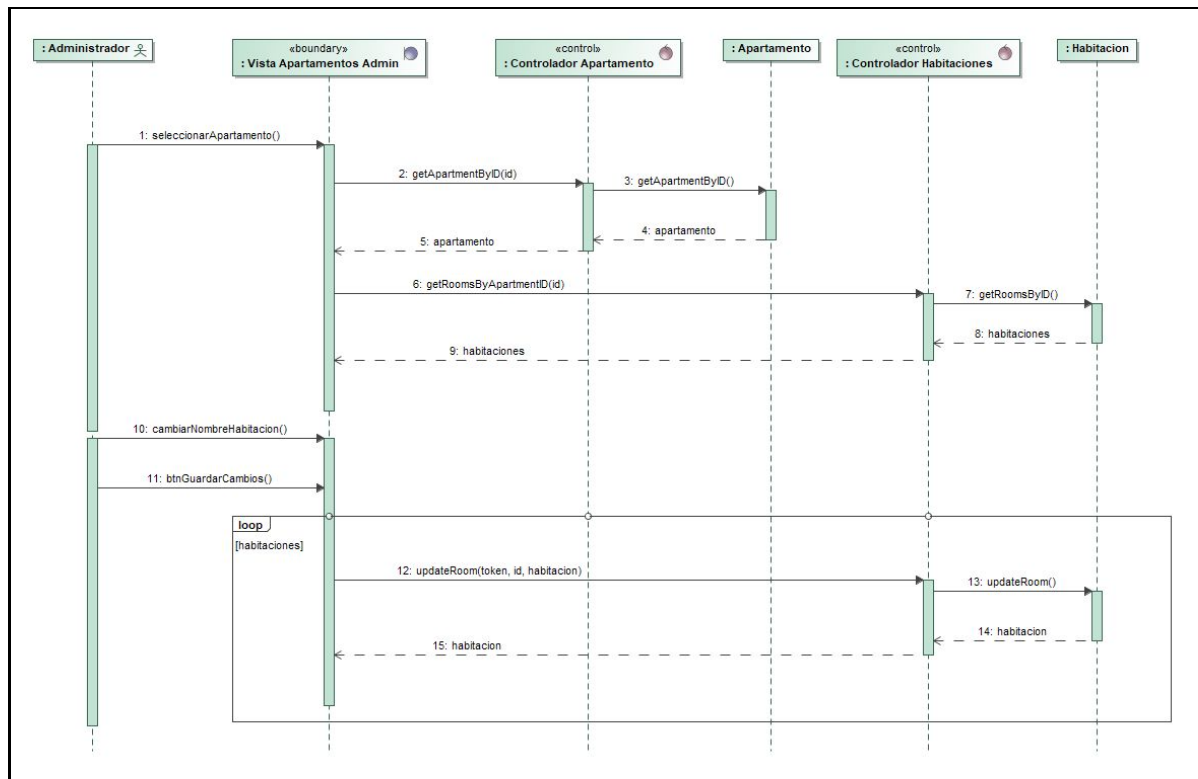
B. Clases de control	navbar.component.ts, panel-admin.component.ts, apartamentos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html apartamentos-admin.component.html

Maquetas de interfaz





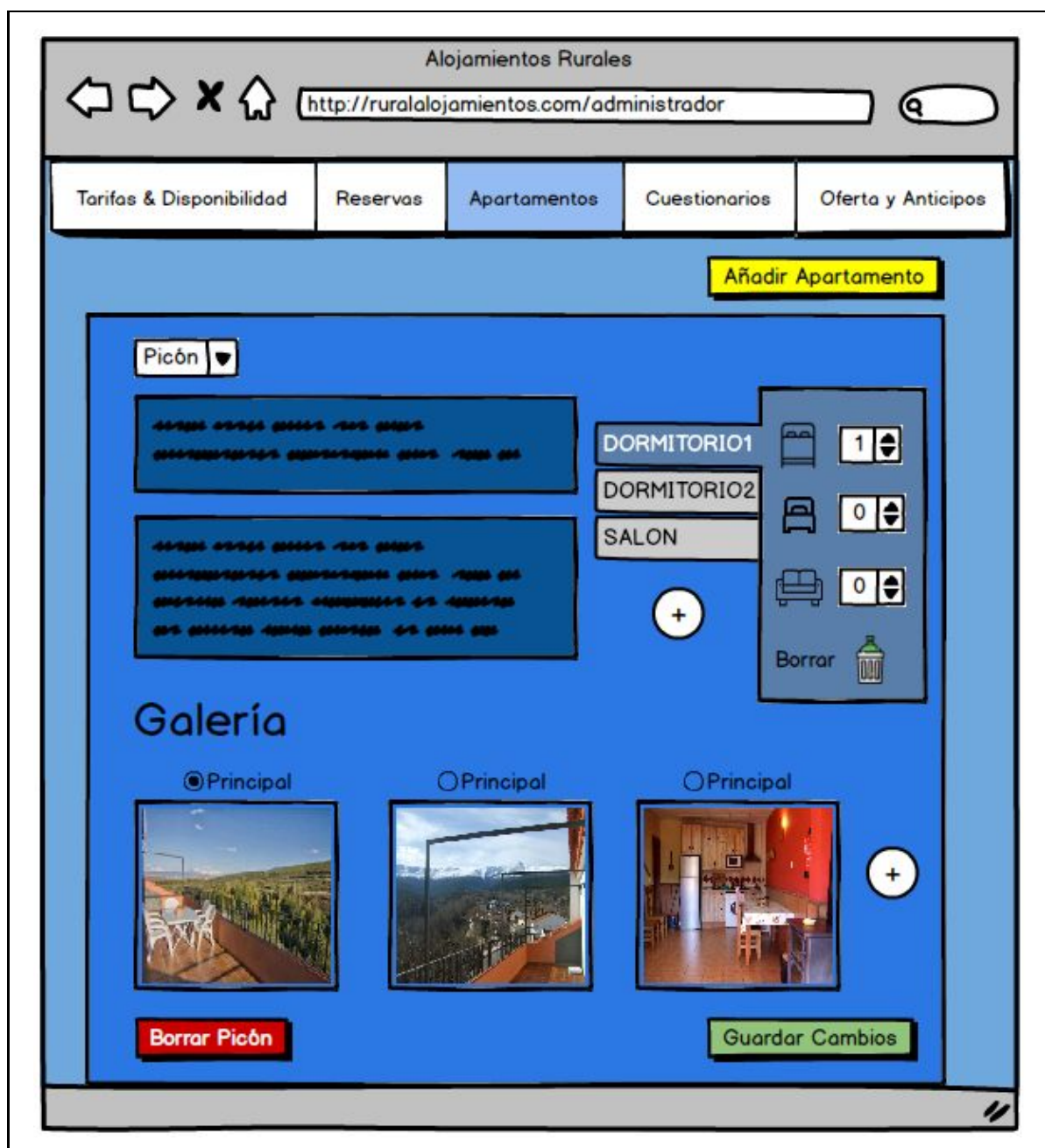
Diagramas de secuencia



4.3.22 Eliminar Habitación

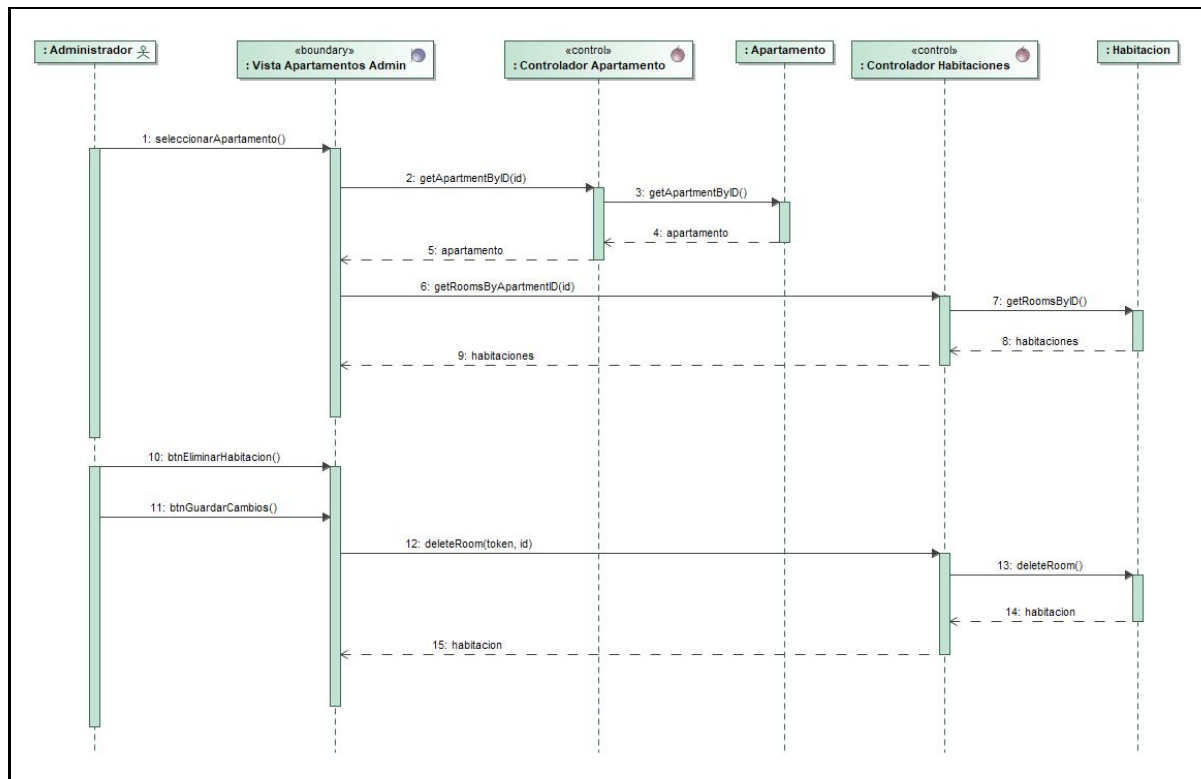
Título	Eliminar Habitación
Descripción	Un usuario administrador modifica un apartamento en su panel de administrador
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema elimina el apartamento en la base de datos
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Apartamentos’ para acceder a la página correspondiente 3. El sistema le ofrece al usuario un scroll para seleccionar el apartamento a visualizar y carga el primero de la lista inicialmente	

<p>4. El usuario presiona el botón de la papelera que se encuentra junto a una habitación y presiona el botón ‘Guardar Cambios’</p> <p>5. El sistema ha eliminado la habitación de la base de datos y muestra un Pop Up</p>	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Habitación
B. Clases de control	navbar.component.ts, panel-admin.component.ts, apartamentos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html apartamentos-admin.component.html
Maquetas de interfaz	





Diagramas de secuencia



4.3.23 Consultar Anticipos

Título	Consultar Anticipos
Descripción	Un usuario administrador visualiza la información de los anticipos en la base de datos
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ofrece al usuario administrador la información de los anticipos existentes
Prioridad	Media
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Oferta & Anticipos’ para acceder a la página correspondiente 3. El sistema ofrece al usuario administrador la información de los anticipos existentes	

Clases de análisis	
A. Clases de entidad	Usuario, Oferta
B. Clases de control	navbar.component.ts, panel-admin.component.ts, oferta-anticipos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html oferta-anticipos-admin.component.html
Maquetas de interfaz	

Alojamientos Rurales

← → ✕ 🏠

q

Tarifas & Disponibilidad
Reservas
Apartamentos
Cuestionarios
Oferta y Anticipos

Título:

Texto:

☐ No hay ninguna oferta actualmente

Guardar Cambios

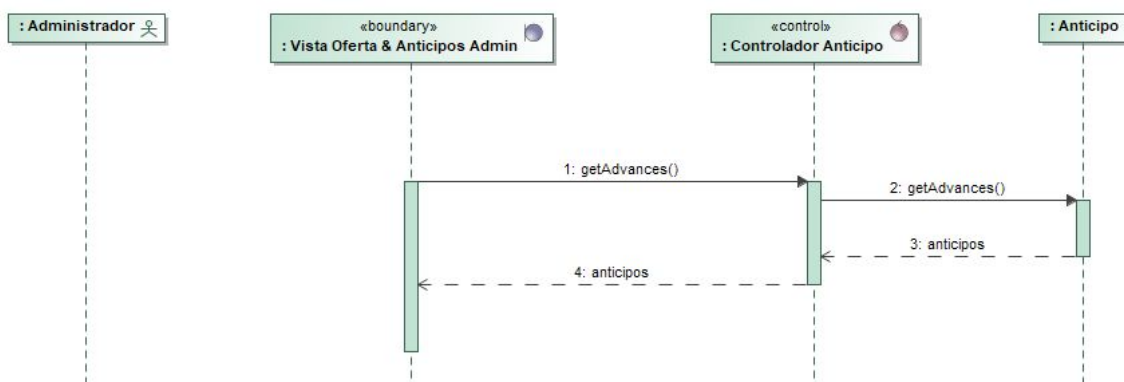
Fechas con Anticipo		
Fechas de Inicio	Fecha de Fin	Eliminar
21/12/2019	09/01/2020	✕

Desde 📅

Hasta 📅

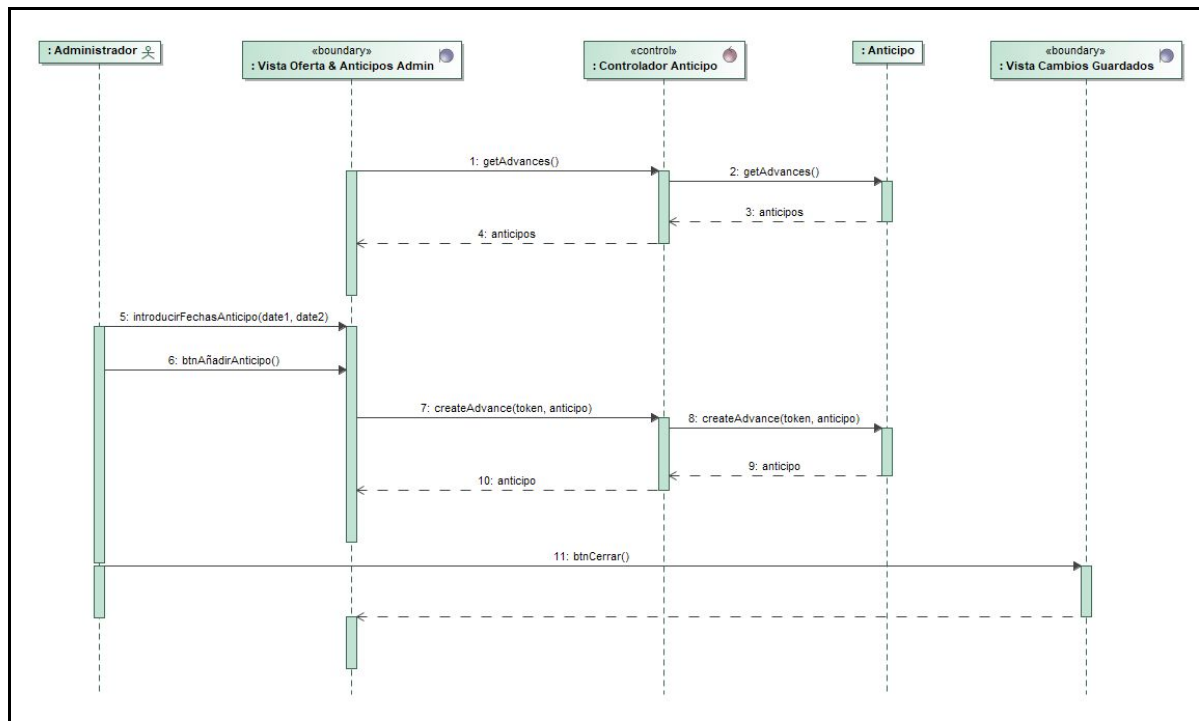
Añadir

Diagramas de secuencia



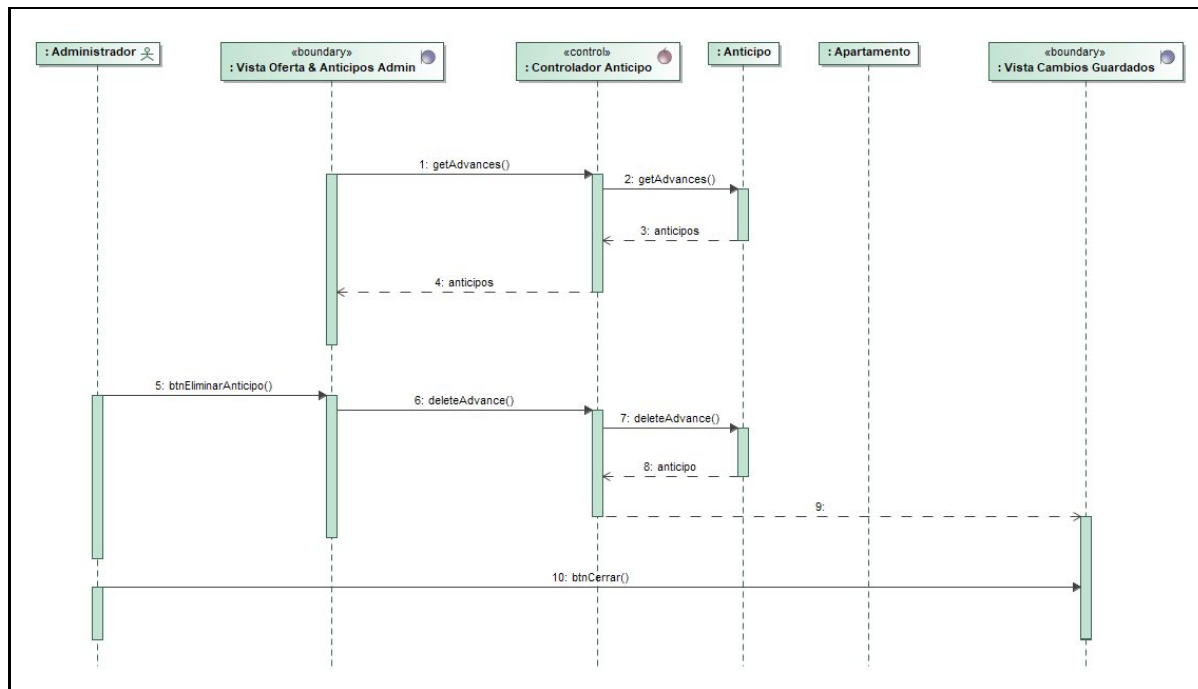
4.3.24 Crear Anticipo

Título	Crear Anticipo
Descripción	Un usuario administrador crea un anticipo
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha creado en la base de datos un nuevo objeto de la clase anticipo
Prioridad	Media
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
<ol style="list-style-type: none">1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación2. El usuario presiona el botón ‘Oferta & Anticipos’ para acceder a la página correspondiente3. El usuario introduce la fecha de inicio y la fecha de fin del nuevo anticipo y presiona el botón ‘Añadir Anticipo’4. El nuevo anticipo ha sido creado en la base de datos y se muestra un Pop Up	
Clases de análisis	
A. Clases de entidad	Usuario, Oferta
B. Clases de control	navbar.component.ts, panel-admin.component.ts, oferta-anticipos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html oferta-anticipos-admin.component.html
Maquetas de interfaz	



4.3.25 Eliminar Anticipo

Título	Eliminar Anticipo
Descripción	Un usuario administrador elimina un anticipo
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha eliminado un anticipo de la base de datos
Prioridad	Media
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Oferta & Anticipos’ para acceder a la página correspondiente 3. El usuario presiona el botón de la papelera en la tabla de anticipos para eliminar uno de ellos 4. Se ha eliminado el anticipo de la base de datos y se muestra un Pop Up	
Clases de análisis	
A. Clases de entidad	Usuario, Oferta
B. Clases de control	navbar.component.ts, panel-admin.component.ts, oferta-anticipos-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html oferta-anticipos-admin.component.html
Maquetas de interfaz	



4.3.26 Consultar Tarifas

Título	Consultar Tarifas
Descripción	Un usuario administrador accede a las tarifas de un apartamento
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha mostrado al usuario las tarifas del apartamento concreto
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Tarifas’ para acceder a la página correspondiente 3. El usuario selecciona el apartamento del cual quiere ver las tarifas 5. Se muestran por pantalla las tarifas de dicho apartamento	
Clases de análisis	

A. Clases de entidad	Usuario, Tarifa, Apartamento
B. Clases de control	navbar.component.ts, panel-admin.component.ts, tarifas-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html tarifas-admin.component.html

Maquetas de interfaz

Alojamientos Rurales

http://ruralalojamientos.com/administrador

Tarifas & Disponibilidad Reservas Apartamentos Cuestionarios Oferta y Anticipos

Apartamento: Picón Cerrar Sesión

2019	Octubre												
Tarifas / Días	1	2	3	4	5	6	7	8	9	10	11	12	>
1N	15	15	15	15	18	18	18	15	15	15	15	15	
2N	18	18	18	18	22	22	22	18	18	18	18	18	
Ocupados		X		X	X					X	X		

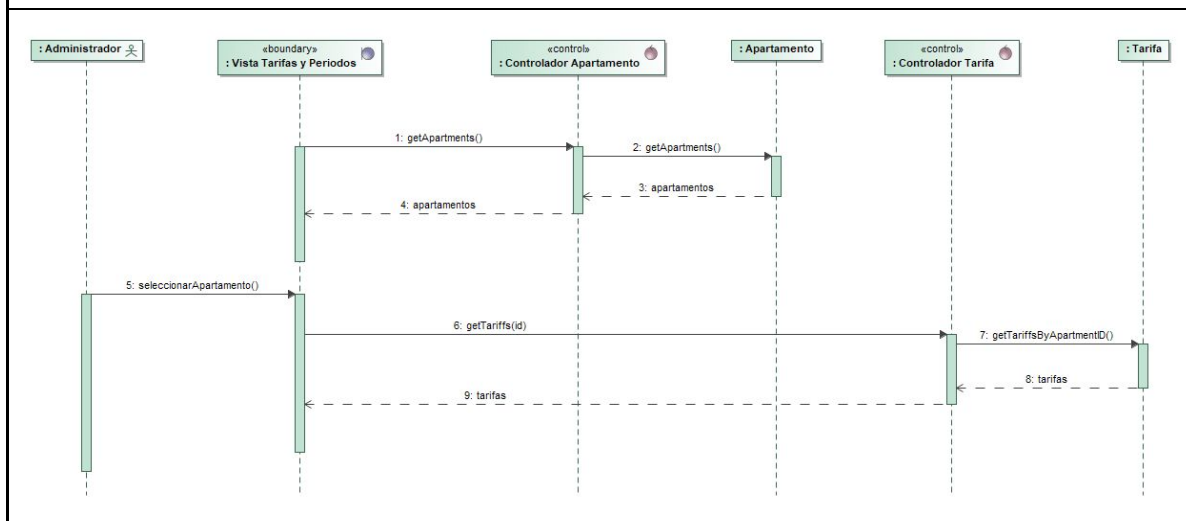
Crear Tarifa:

Noches: 3 Precio por noche: 23 Añadir

Eliminar Tarifa:

Noches: 2 Eliminar

Diagramas de secuencia



4.3.27 Crear Tarifa

Título	Crear Tarifa
Descripción	Un usuario administrador crea una tarifa
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha creado en la base de datos un nuevo objeto de la clase tarifa
Prioridad	Media
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020

Escenario principal

1. El usuario accede a la web conectado y presiona el botón 'Panel de Administrador' de la barra de navegación
2. El usuario presiona el botón 'Tarifas' para acceder a la página correspondiente
3. El usuario selecciona el apartamento del cual quiere crear una nueva tarifa
4. El usuario introduce el número de noches, el precio diario y el precio de fin de semana y pulsa en el botón Añadir
5. Se ha creado una nueva tarifa para ese apartamento en la base de datos y se muestra un Pop Up

Clases de análisis

A. Clases de entidad

Usuario, Tarifa, Apartamento

B. Clases de control	navbar.component.ts, panel-admin.component.ts, tarifas-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html tarifas-admin.component.html

Maquetas de interfaz

Alojamientos Rurales

← → ✕ 🏠

http://ruralalojamientos.com/administrador

🔍

Tarifas & Disponibilidad

Reservas

Apartamentos

Cuestionarios

Oferta y Anticipos

Apartamento:

Picón ▼

Cerrar Sesión

2019 ▼

Octubre ▼

Tarifas / Días	1	2	3	4	5	6	7	8	9	10	11	12	>
1N	15	15	15	15	18	18	18	15	15	15	15	15	
2N	18	18	18	18	22	22	22	18	18	18	18	18	
Ocupados		✕		✕	✕					✕	✕		

Crear Tarifa:

Noches: 3 ▼

Precio por noche: 23 ▼

Añadir

Eliminar Tarifa:

Noches: 2 ▼

Eliminar

Alojamientos Rurales

← → ✕ 🏠

Tarifas & Disponibilidad
Reservas
Apartamentos
Cuestionarios
Oferta y Anticipos

Apartamento: Picón ▼

2019 ▼
Octubre ▼

Tarifas / Días	1	2	3	4	5	6	7	8	9	10	11	12	>
1N	15	15	15	15	18	18	18	15	15	15	15	15	
2N	18	18	18	18	22	22	22	18	18	18	18	18	
3N	23	23	23	23	23	23	23	23	23	23	23	23	
Ocupados		✕		✕	✕					✕	✕		

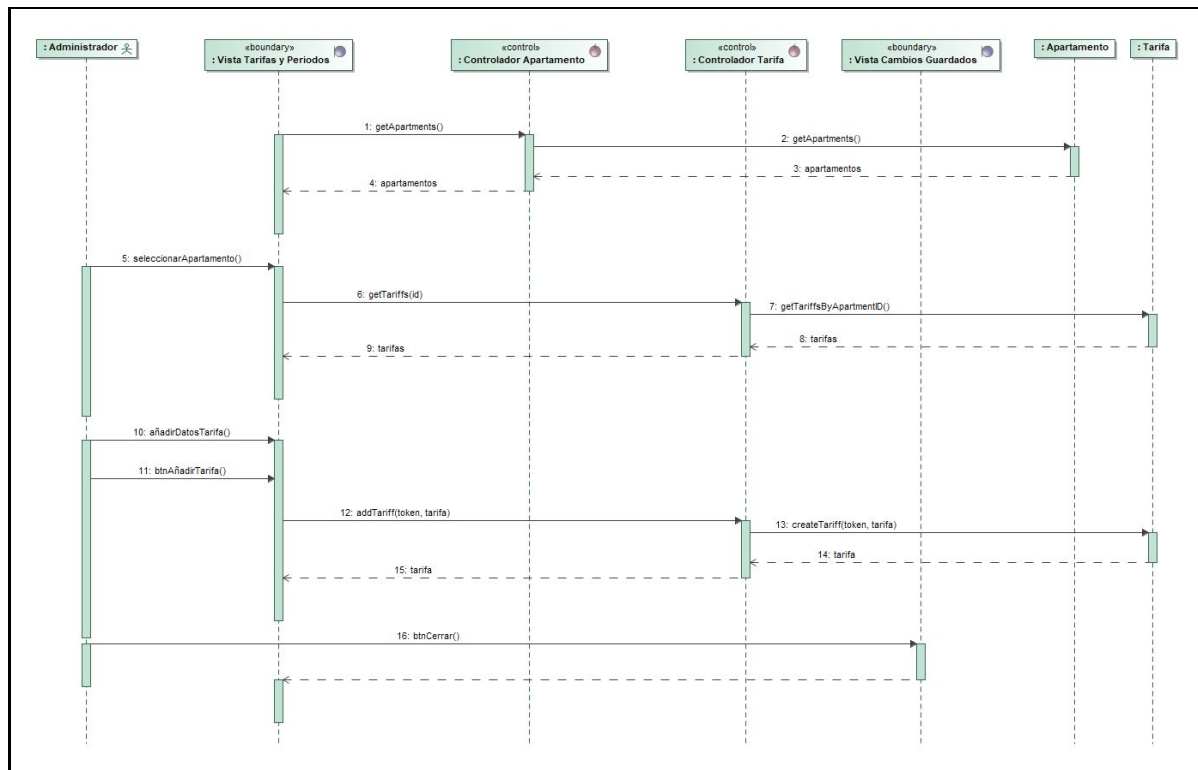
Crear Tarifa:

Noches: 3 ▼
Precio por noche: 23 ▼
Añadir

Eliminar Tarifa:

Noches: 3 ▼
Eliminar

Diagramas de secuencia



4.3.28 Eliminar Tarifa

Título	Eliminar Tarifa
Descripción	Un usuario administrador elimina una tarifa
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha eliminado un objeto de la clase tarifa de la base de datos
Prioridad	Media
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Tarifas’ para acceder a la página correspondiente 3. El usuario selecciona un apartamento para ver sus tarifas 4. El usuario presiona el botón de la papelera de una tarifa de la tabla de tarifas 5. La tarifa se elimina correctamente de la base de datos y se muestra un Pop Up 	

Clases de análisis	
A. Clases de entidad	Usuario, Tarifa, Apartamento
B. Clases de control	navbar.component.ts, panel-admin.component.ts, tarifas-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html tarifas-admin.component.html
Maquetas de interfaz	

Alojamientos Rurales

← → ✕ 🏠

Tarifas & Disponibilidad
Reservas
Apartamentos
Cuestionarios
Oferta y Anticipos

Apartamento: Picón ▼

2019 ▼
Octubre ▼

Tarifas / Días	1	2	3	4	5	6	7	8	9	10	11	12	>
1N	15	15	15	15	18	18	18	15	15	15	15	15	
2N	18	18	18	18	22	22	22	18	18	18	18	18	
3N	23	23	23	23	23	23	23	23	23	23	23	23	
Ocupados		✕		✕	✕					✕	✕		

Crear Tarifa:

Noches: 3 ▼

Precio por noche: 23 ▼

Añadir

Eliminar Tarifa:

Noches: 3 ▼

Eliminar

Alojamientos Rurales

←
→
✕
🏠

http://ruralalojamientos.com/administrador

Q

Tarifas & Disponibilidad
Reservas
Apartamentos
Cuestionarios
Oferta y Anticipos

Apartamento: Picón ▼
Cerrar Sesión

2019 ▼	Octubre ▼												
Tarifas / Días	1	2	3	4	5	6	7	8	9	10	11	12	>
1N	15	15	15	15	18	18	18	15	15	15	15	15	
2N	18	18	18	18	22	22	22	18	18	18	18	18	
Ocupados		✕		✕	✕					✕	✕		

Crear Tarifa:

Noches: 3 ▼

Precio por noche: 23 ▼

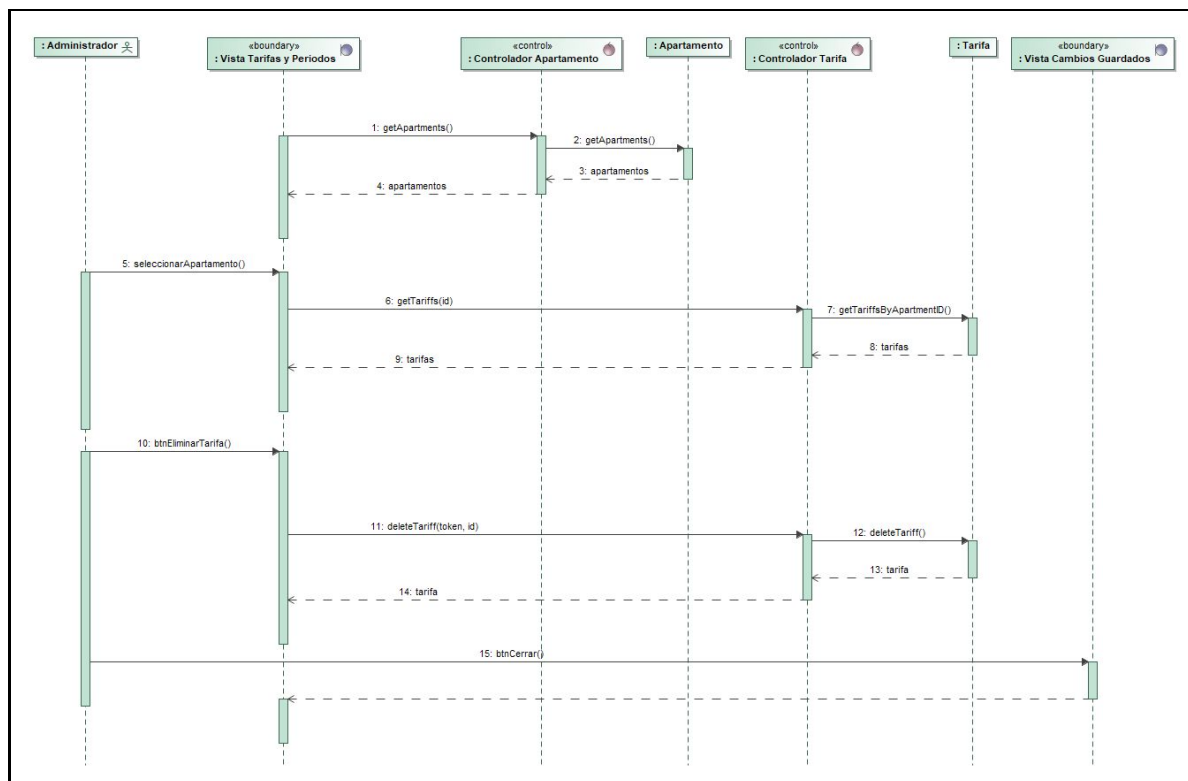
Añadir

Eliminar Tarifa:

Noches: 2 ▼

Eliminar

Diagramas de secuencia



4.3.29 Consultar Periodos

Título	Consultar Periodos
Descripción	Un usuario administrador accede a los periodos de una tarifa de un apartamento
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha mostrado al usuario los periodos de una tarifa
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Tarifas’ para acceder a la página correspondiente 3. El usuario selecciona el apartamento del cual quiere ver las tarifas 5. El usuario selecciona de qué tarifa quiere ver los periodos	

6. El sistema muestra al usuario los periodos de la tarifa seleccionada

Clases de análisis

A. Clases de entidad

Usuario, Tarifa, Apartamento, Periodo

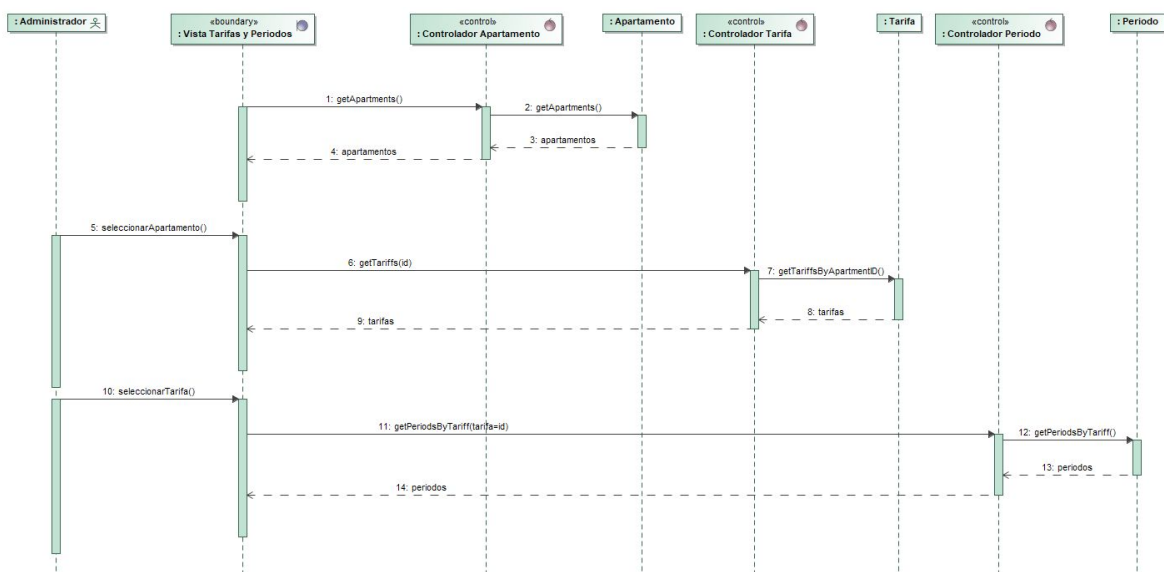
B. Clases de control

navbar.component.ts,
panel-admin.component.ts,
tarifas-admin.component.ts

C. Clases de interfaz

navbar.component.html,
panel-admin.component.html
tarifas-admin.component.html

Diagramas de secuencia



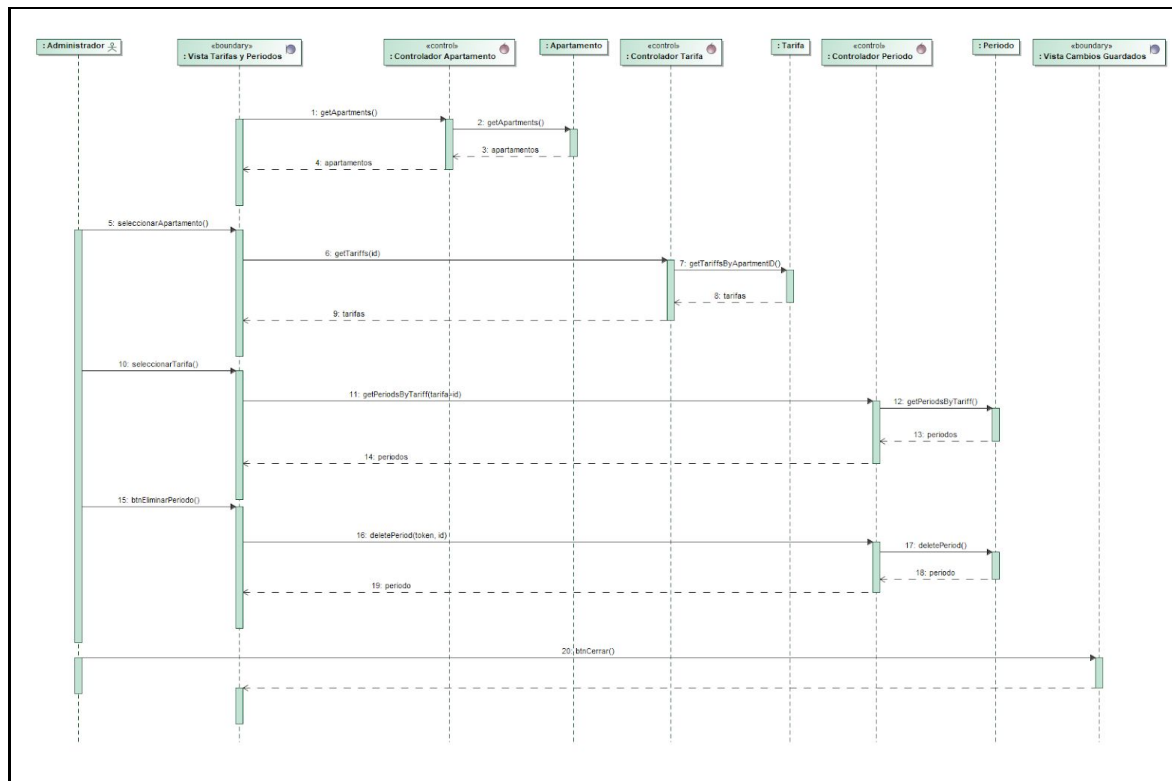
4.3.30 Añadir Periodo

Título	Añadir Periodo
Descripción	Un usuario administrador crea un nuevo periodo para una tarifa
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha creado un objeto de la clase periodo en la base de datos
Prioridad	Alta
Autor(es)	José María Gómez García

Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Tarifas’ para acceder a la página correspondiente 3. El usuario selecciona el apartamento del cual quiere ver las tarifas 5. El usuario selecciona de qué tarifa quiere ver los periodos 6. El usuario introduce la fecha de comienzo y fin además de la suma del nuevo periodo y presiona el botón Añadir Periodo 7. El nuevo periodo se ha creado en la base de datos 	
Clases de análisis	
A. Clases de entidad	Usuario, Tarifa, Apartamento, Periodo
B. Clases de control	navbar.component.ts, panel-admin.component.ts, tarifas-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html tarifas-admin.component.html
Diagramas de secuencia	
<pre> sequenceDiagram actor Admin as :Administrador participant Vista as :Vista Tarifas y Periodos participant ControlApto as :Controlador Apartamento participant ControlTarifa as :Controlador Tarifa participant ControlPeriodo as :Controlador Periodo participant Tarifa as :Tarifa participant Periodo as :Periodo participant Apartamento as :Apartamento participant VistaGuardados as :Vista Cambios Guardados Admin->>Vista: 1. getApartments() activate Vista Vista->>ControlApto: 2. getApartments() activate ControlApto ControlApto-->>Vista: 3. apartamentos deactivate ControlApto Vista-->>Admin: 4. apartamentos deactivate Vista Admin->>Vista: 5. seleccionarApartamento() activate Vista Vista->>ControlTarifa: 6. getTariffs(id) activate ControlTarifa ControlTarifa->>Tarifa: 7. getTariffsBy(ApartmentID) activate Tarifa Tarifa-->>ControlTarifa: 8. tarifas deactivate Tarifa ControlTarifa-->>Vista: 9. tarifas deactivate ControlTarifa Vista-->>Admin: 10. seleccionarTarifa() deactivate Vista Admin->>Vista: 11. getPeriodsBy(Tariff/tarifa+id) activate Vista Vista->>ControlPeriodo: 12. getPeriodsBy(Tariff) activate ControlPeriodo ControlPeriodo->>Periodo: 13. periodos activate Periodo Periodo-->>ControlPeriodo: 14. periodos deactivate Periodo ControlPeriodo-->>Vista: 15. periodos deactivate ControlPeriodo Vista-->>Admin: 16. añadirDatosPeriodo() deactivate Vista Admin->>Vista: 17. btnAñadirPeriodo() activate Vista Vista->>ControlPeriodo: 18. createPeriod(token, periodo) activate ControlPeriodo ControlPeriodo->>Periodo: 19. createPeriod(token, periodo) activate Periodo Periodo-->>ControlPeriodo: 20. periodo deactivate Periodo ControlPeriodo-->>Vista: 21. btnCerrar() deactivate ControlPeriodo Vista-->>Admin: 22. btnCerrar() deactivate Vista deactivate Admin </pre>	

4.3.31 Eliminar Periodo

Título	Eliminar Periodo	
Descripción	Un usuario administrador elimina un periodo de la base de datos	
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador	
Post-condición	El sistema ha eliminado un objeto de la clase periodo en la base de datos	
Prioridad	Alta	
Autor(es)	José María Gómez García	
Control de cambios	Última modificación: 7 de Septiembre de 2020	
Escenario principal		
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Tarifas’ para acceder a la página correspondiente 3. El usuario selecciona el apartamento del cual quiere ver las tarifas 5. El usuario selecciona de qué tarifa quiere ver los periodos 6. El usuario presiona el botón de la papelera de uno de los periodos de la tarifa 7. El periodo se elimina de la base de datos y muestra un Pop Up		
Clases de análisis		
A. Clases de entidad	Usuario, Tarifa, Apartamento, Periodo	
B. Clases de control	navbar.component.ts, panel-admin.component.ts, tarifas-admin.component.ts	
C. Clases de interfaz	navbar.component.html, panel-admin.component.html tarifas-admin.component.html	
Diagramas de secuencia		



4.3.32 Crear Ocupado

Título	Crear Ocupado
Descripción	Un usuario administrador crea un nuevo objeto ocupado en la base de datos para indicar que dicho día, un apartamento concreto está ocupado
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha creado un objeto de la clase ocupado en la base de datos
Prioridad	Alta
Autor(es)	José María Gómez García
Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
<ol style="list-style-type: none"> 1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Disponibilidad’ para acceder a la página correspondiente 	

3. El usuario selecciona el año y el mes para el cual quiere cargar los días ocupados 4. El sistema carga la tabla y muestra la disponibilidad de los apartamentos para ese mes 5. El usuario presiona uno de los días y crea en la base de datos el objeto de la clase ocupado o lo elimina si ya existe	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Ocupado
B. Clases de control	navbar.component.ts, panel-admin.component.ts, disponibilidad-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html disponibilidad-admin.component.html
Maquetado de Interfaz	

Alojamientos Rurales

← → ✕ 🏠

Tarifas & Disponibilidad
Reservas
Apartamentos
Cuestionarios
Oferta y Anticipos

Apartamento:

Picón ▼

Cerrar Sesión

2019 ▼

Octubre ▼

Tarifas / Días	1	2	3	4	5	6	7	8	9	10	11	12	>
1N	15	15	15	15	18	18	18	15	15	15	15	15	
2N	18	18	18	18	22	22	22	18	18	18	18	18	
Ocupados		✕		✕	✕					✕	✕		

Crear Tarifa:

Noches:

3 ▼

Precio por noche:

23 ▼

Añadir

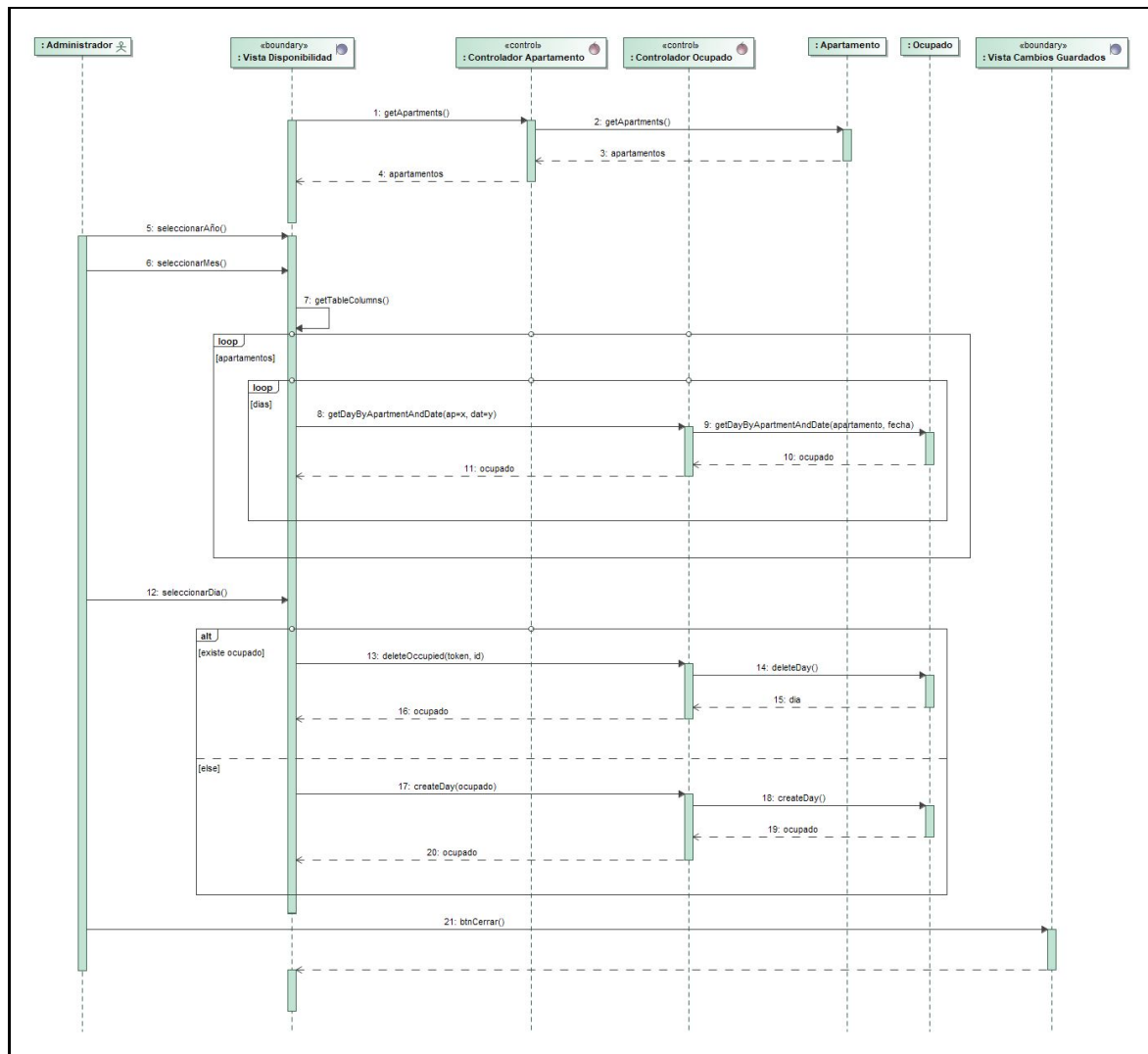
Eliminar Tarifa:

Noches:

2 ▼

Eliminar

Diagramas de Secuencia



4.3.33 Eliminar Ocupado

Título	Eliminar Ocupado
Descripción	Un usuario administrador elimina un objeto ocupado en la base de datos para indicar que dicho día va a pasar a estar libre para ocuparlo
Pre-condición	Estar registrado, conectado en la aplicación y tener el rol de administrador
Post-condición	El sistema ha eliminado un objeto de la clase ocupado en la base de datos
Prioridad	Alta
Autor(es)	José María Gómez García

Control de cambios	Última modificación: 7 de Septiembre de 2020
Escenario principal	
1. El usuario accede a la web conectado y presiona el botón ‘Panel de Administrador’ de la barra de navegación 2. El usuario presiona el botón ‘Disponibilidad’ para acceder a la página correspondiente 3. El usuario selecciona el año y el mes para el cual quiere cargar los días ocupados 4. El sistema carga la tabla y muestra la disponibilidad de los apartamentos para ese mes 5. El usuario presiona uno de los días y crea en la base de datos el objeto de la clase ocupado o lo elimina si ya existe	
Clases de análisis	
A. Clases de entidad	Usuario, Apartamento, Ocupado
B. Clases de control	navbar.component.ts, panel-admin.component.ts, disponibilidad-admin.component.ts
C. Clases de interfaz	navbar.component.html, panel-admin.component.html disponibilidad-admin.component.html
Maquetado de Interfaz	

Alojamientos Rurales

← → ✕ 🏠

Tarifas & Disponibilidad
Reservas
Apartamentos
Cuestionarios
Oferta y Anticipos

Apartamento: Picón ▼
Cerrar Sesión

2019 ▼	Octubre ▼												
Tarifas / Días	1	2	3	4	5	6	7	8	9	10	11	12	>
1N	15	15	15	15	18	18	18	15	15	15	15	15	
2N	18	18	18	18	22	22	22	18	18	18	18	18	
Ocupados		✕		✕	✕					✕	✕		

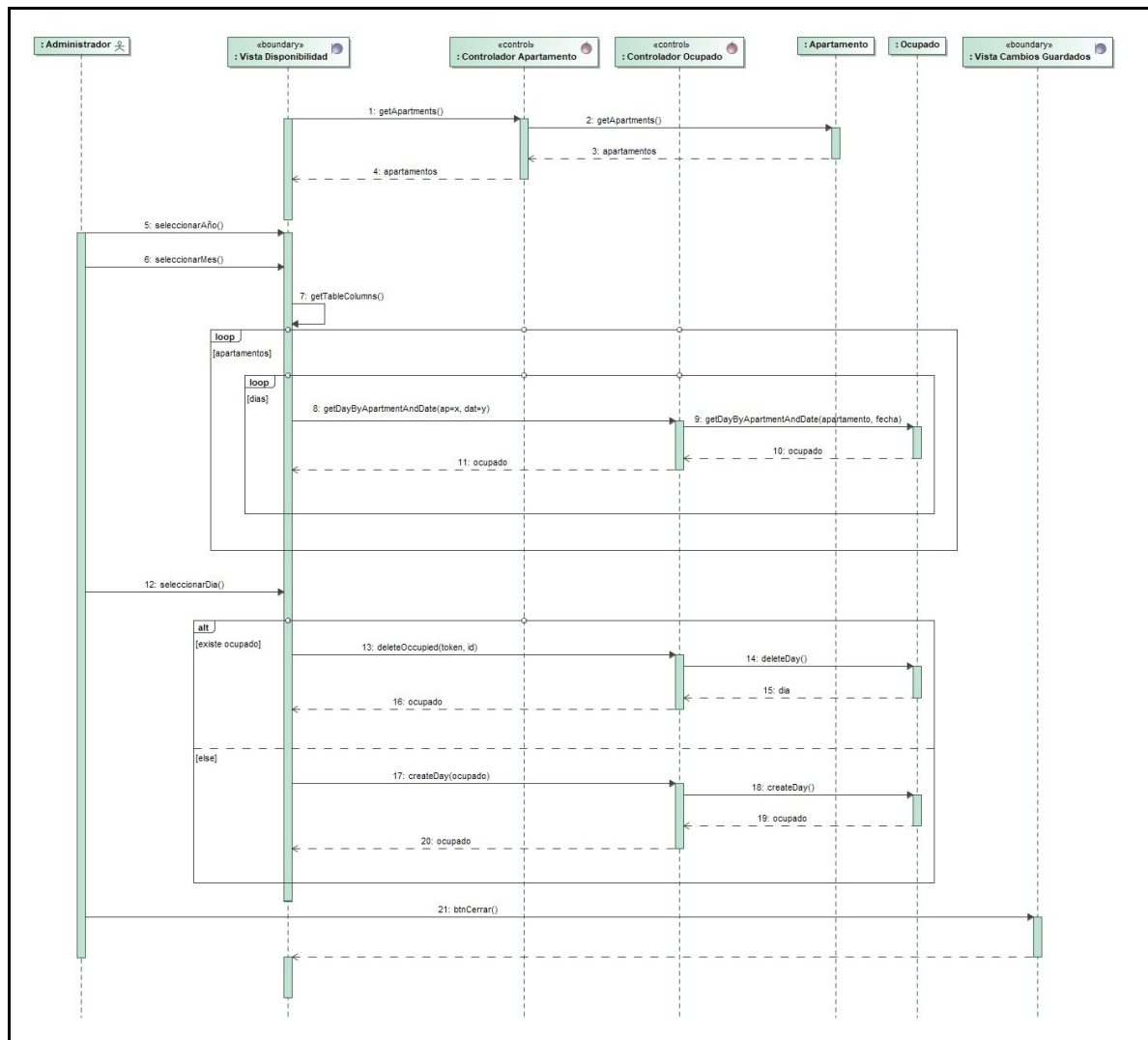
Crear Tarifa:

Noches: 3 ▼ Precio por noche: 23 ▼ Añadir

Eliminar Tarifa:

Noches: 2 ▼ Eliminar

Diagramas de Secuencia



5

Arquitectura y Entorno Tecnológico

En este apartado vamos a usar un diagrama para describir el funcionamiento del sistema (Figura 29) mediante un esquema. Además, explicaremos un poco cada una de las herramientas software que serán utilizadas para la elaboración de la aplicación web.

5.1 Arquitectura

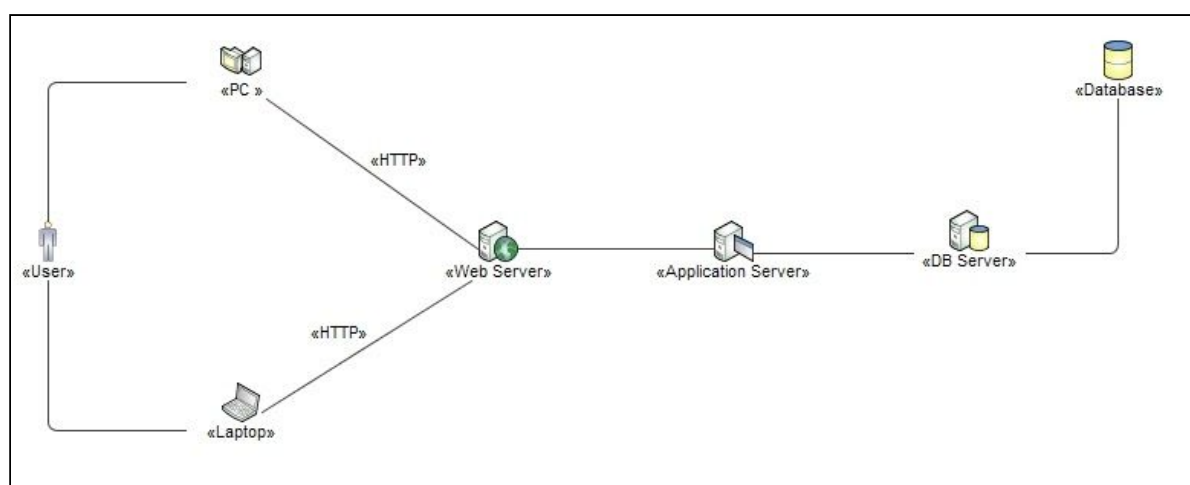


Figura 29: Diagrama de arquitectura en MagicDraw

El usuario accede a través de un navegador, ya sea en un ordenador o laptop donde se esté corriendo el sistema, al servidor web en modo local.

Ejemplo de dirección web para acceder: <http://localhost:4200/>

Siendo 4200 el puerto donde se ha montado la aplicación web.

Accede al servidor de la aplicación y comienza a usar la web utilizando los componentes de la misma. El servidor de aplicación realiza peticiones a la API, comunicándose así con el servidor de la base de datos que accede a la bd y ejecuta las sentencias correspondientes.

5.2 Entorno Tecnológico

A continuación vamos a analizar brevemente las tecnologías que utilizaré para realizar la implementación de la página web; Visual Studio Code, Mongo DB, Postman, Angular, Mongoose & Connect Multiparty.

5.2.1 Visual Studio Code



Figura 30: Logo de Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Es personalizable y compatible con varios lenguajes de programación (Figura 31), basta con descargar una extensión para el lenguaje que necesitas (Angular o Typescript, por ejemplo) y el sistema se encarga de ofrecer colores para entender mejor el código y detectar errores, si has escrito algo de forma errónea.

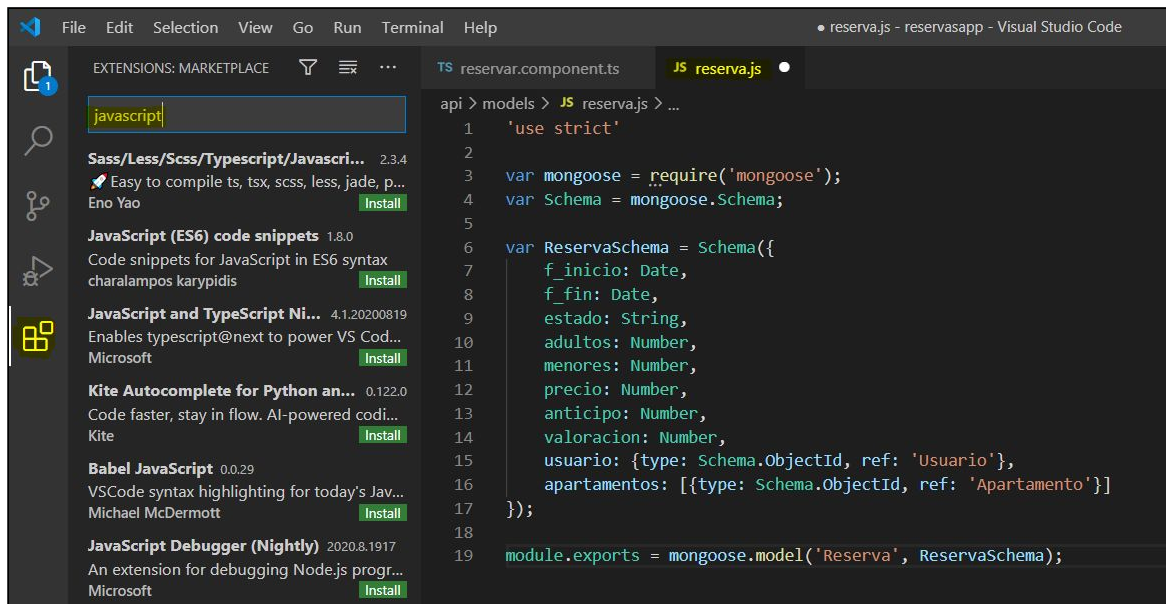


Figura 31: Buscador de extensiones de Visual Studio Code

En el apartado Extensiones, basta con escribir el lenguaje deseado e instalarlo, y el sistema se encargará de detectar fallos en todos los archivos cuya extensión corresponda. Por ejemplo, en este caso, tenemos la extensión javascript y a la derecha un ejemplo de un archivo con extensión js.

Además, el sistema incluye un terminal (Figura 32) en el que puedes ejecutar comandos, por ejemplo para correr la aplicación en un servidor local, accediendo al directorio en el que se encuentra el proyecto.

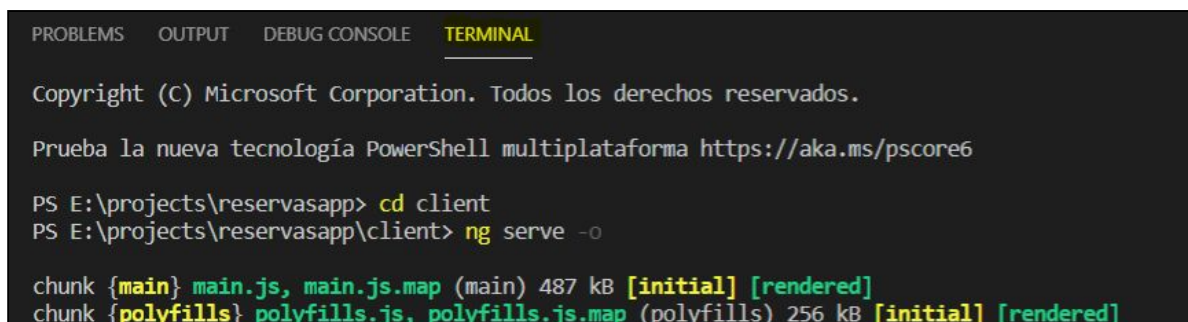


Figura 32: Terminal de Visual Studio Code

El sistema te permite abrir un directorio y navegar fácilmente por los archivos (Figura 33), de manera que puedes trabajar con todo el código en una misma ventana sin perderte, pues además el sistema adjudica iconos a cada tipo de archivo.

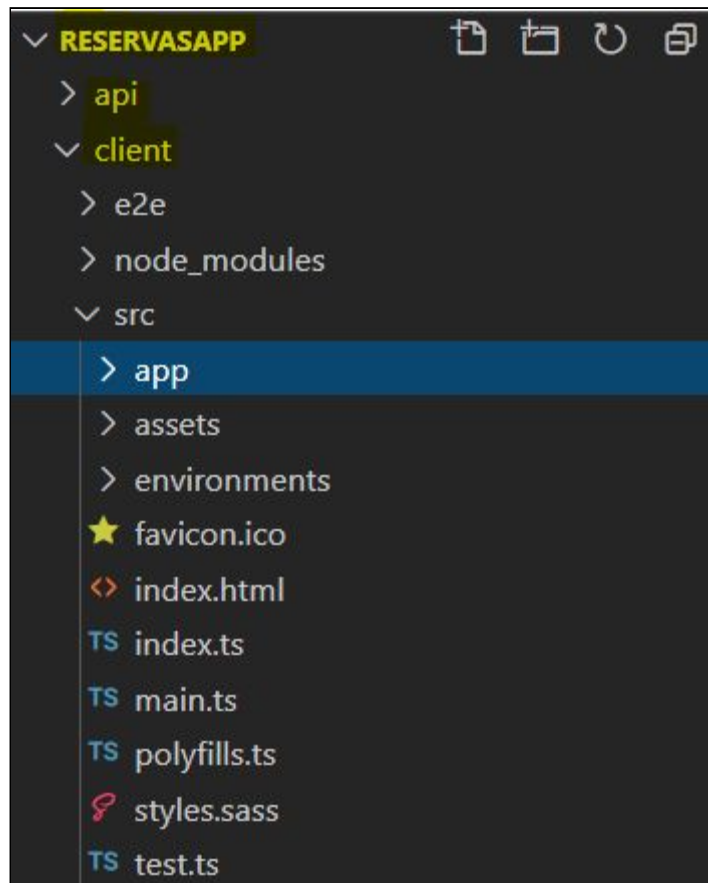


Figura 33: Navegador de archivos de Visual Studio Code

Al estar aprendiendo nuevos lenguajes de desarrollo, el sistema detecta fallos de compilación (Figura 34) que ayudan a manejarse con mayor soltura.

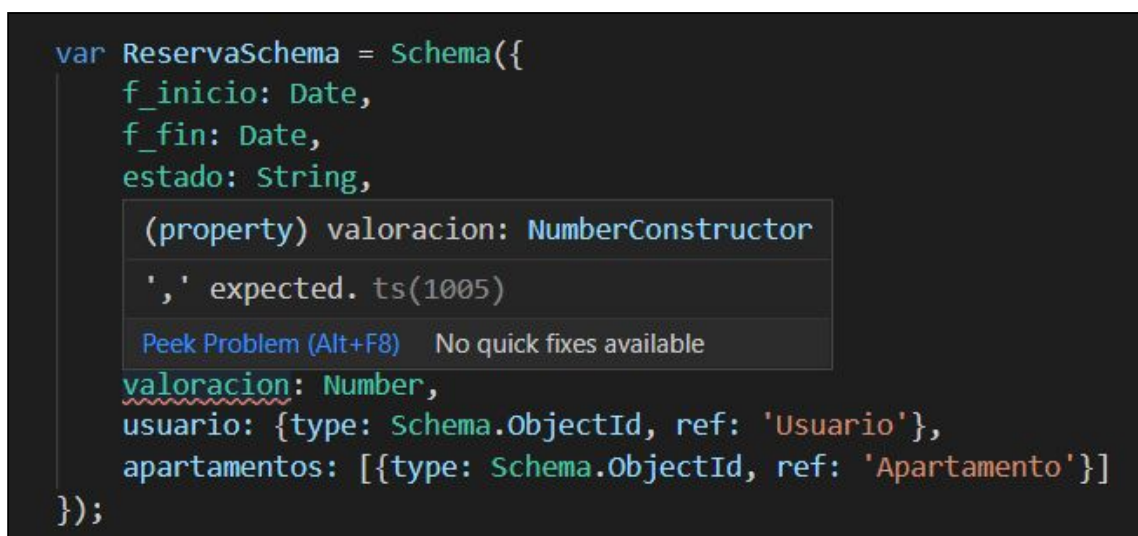


Figura 34: Aviso de error de compilación en Visual Studio Code

5.2.2 Mongo DB



Figura 35: Logo de Mongo DB

MongoDB es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube.

Se trata de una base de datos no relacional o también denominada No-SQL. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad) y habitualmente escalan bien horizontalmente. Los sistemas NoSQL se denominan a veces "no solo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL.

MongoDB es una base de datos documental, lo que significa que almacena datos en forma de documentos tipo JSON (Figura 36). Frente al tradicional modelo de filas y columnas, esta es más expresiva y potente y proporciona una mejor forma de concebir los datos.

```
{
  Nombre: "Miguel",
  Apellidos: "Parada",
  Edad: 39,
  Aficiones: ["Música", "Ciclismo", "Baloncesto"],
  Amigos: [
    {
      Nombre: "Marie",
      Edad: 35
    },
    {
      Nombre: "Elsa",
      Edad: 42
    }
  ]
}
```

Figura 36: Objeto en formato JSON

Esto nos permite trabajar con objetos tal y como lo hemos hecho en la asignatura Programación Orientada a Objetos.

Incluye un potente lenguaje de consulta rico y expresivo (Figura 37) que permite filtrar y ordenar por cualquier campo, independientemente de cómo esté incrustado en un documento.



```
> db.users.find({ "address.zip" : "90404" })
{ "_id": "5cf0029caff5056591b0ce7d", "firstname": "Jane", "lastname": "Smith", "address": { "zip": "90404", "city": "Miami", "street": "123 Main St" } }
{ "_id": "507f1f77bcf86cd799439011", "firstname": "Jon", "lastname": "Doe", "address": { "zip": "90404", "city": "Miami", "street": "456 Main St" } }
{ "_id": "5349b4ddd2781d08c09890f3", "firstname": "Jim", "lastname": "Brown", "address": { "zip": "90404", "city": "Miami", "street": "789 Main St" } }
{ "_id": "5bf142459b72e12b2b1b2cd", "firstname": "Jeff", "lastname": "Green", "address": { "zip": "90404", "city": "Miami", "street": "101 Main St" } }
{ "_id": "5cf003283b23d04a40d5f88a", "firstname": "Jerry", "lastname": "White", "address": { "zip": "90404", "city": "Miami", "street": "202 Main St" } }
{ "_id": "5bf142459b72e12b2b1b2cd", "firstname": "Jai", "lastname": "Black", "address": { "zip": "90404", "city": "Miami", "street": "303 Main St" } }
{ "_id": "5cf0036deaa1742dd225ea35", "firstname": "Jess", "lastname": "Blue", "address": { "zip": "90404", "city": "Miami", "street": "404 Main St" } }
{ "_id": "54495ad94c934721ede76d90", "firstname": "Jill", "lastname": "Pink", "address": { "zip": "90404", "city": "Miami", "street": "505 Main St" } }
{ "_id": "566eb3c704c7b31facbb0007", "firstname": "Janet", "lastname": "Red", "address": { "zip": "90404", "city": "Miami", "street": "606 Main St" } }
{ "_id": "5a999cc461d36489a27f2563", "firstname": "Jan", "lastname": "Purple", "address": { "zip": "90404", "city": "Miami", "street": "707 Main St" } }
```

Figura 37: Consulta en Mongo DB

Además, Mongo dispone de una herramienta visual llamada Mongo Compass que permite buscar, visualizar y trabajar con los datos en una interfaz de uso intuitiva, así como manipular datos manualmente (Figura 38).

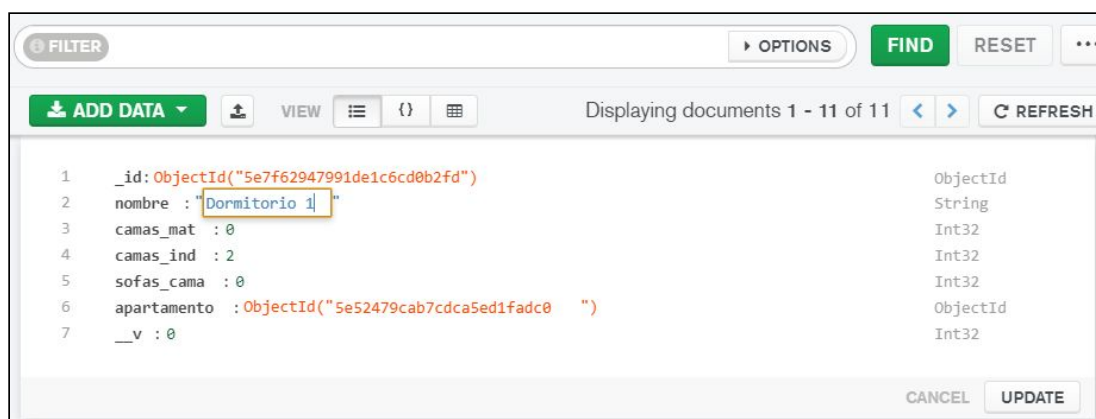


Figura 38: Modificación de la base de datos en Mongo Compass

Estas son algunas de las características que hacen a MongoDB una de las mejores opciones en el mercado:

- ❑ Da respuesta a la necesidad de almacenamiento de todo tipo de datos: estructurados, semi estructurados y no estructurados.
- ❑ Tiene un gran rendimiento en cuanto a escalabilidad y procesado de la información.
- ❑ Puede procesar la gran cantidad de información que se genera hoy en día (millones de usuarios utilizando las mismas aplicaciones, redes sociales que crecen sin parar, internet de las cosas, cloud computing, etc).
- ❑ Se adapta a las necesidades actuales de las aplicaciones (millones de usuarios, miles de peticiones por segundo, etc).
- ❑ Da a los desarrolladores todas las funcionalidades que tienen las bases de datos relacionales (como índices sobre campos secundarios, un completo lenguaje para realizar las consultas, etc).
- ❑ Seguridad de autenticación, autorización, etc

5.2.3 Angular



Figura 39: Logo de Angular

Angular es un framework Javascript potente, muy adecuado para el desarrollo de aplicaciones frontend modernas, de complejidad media o elevada. El tipo de aplicación Javascript que se desarrolla con Angular es del estilo SPA (Single Page Application).

El desarrollo en Angular se hace por medio de TypeScript (aunque también se podría desarrollar con Javascript, todas las guías y recomendaciones se basan en usar TypeScript), un superset del lenguaje Javascript que ofrece muchas herramientas adicionales al lenguaje, como el tipado estático o los decoradores.

Angular no es adecuado por tanto para todo tipo de proyectos. Por ejemplo, un sitio web con WordPress (por poner un ejemplo claro) no sería un buen candidato para incluir Angular en el stack de tecnologías. Hay que darse cuenta que Angular no es una librería para el desarrollo de componentes dinámicos de interfaz gráfica, sino todo un framework para el desarrollo de una aplicación completa y más o menos compleja.

Para este proyecto he decidido usar Angular puesto que lo que vamos a desarrollar es una Aplicación Web completa desde 0. A continuación nombraremos algunas de las ventajas de usar Angular para tu proyecto:

- ❑ **Ahorro de tiempo:** Cuando empezamos a pensar en cómo crear una aplicación web, hemos de tomar una serie de decisiones; por ejemplo, la arquitectura de la aplicación, su organización, etc. Angular ya lo hace por ti, por lo que puedes centrarte en cosas más importantes, como las funcionalidades de tu aplicación web.
- ❑ **Usa lenguaje TypeScript:** La documentación es más consistente porque la sintaxis y la forma de leer los códigos de la información es siempre la misma.
- ❑ **Mantenibilidad:** Al usar TypeScript, cualquier cambio que deba hacerse en la aplicación podrá llevarse a cabo rápidamente y sin errores.
- ❑ **Hace posible intercambiar o añadir programadores en los proyectos:** Cualquier programador de Angular puede leer el código escrito por otro programador de Angular y comprenderlo muy rápidamente. Esto es una gran ventaja a la hora de trabajar en equipo o de retomar proyectos inacabados.
- ❑ **Utiliza componentes web:** Un componente web es una porción de código que puede reutilizarse en otros proyectos hechos con Angular. Además, son fáciles de convertir en componentes web nativos, por lo que pueden usarse de nuevo en otro tipo de aplicaciones.

A continuación veremos cómo crear un proyecto fácilmente con Angular. Abriendo desde Visual Studio Code un nuevo terminal, basta con situarnos en el directorio en el que queremos que se genere nuestra aplicación y ejecutar el comando **ng serve my-app-name**. El sistema creará todos los archivos necesarios y dependencias para empezar a programar (Figura 40).

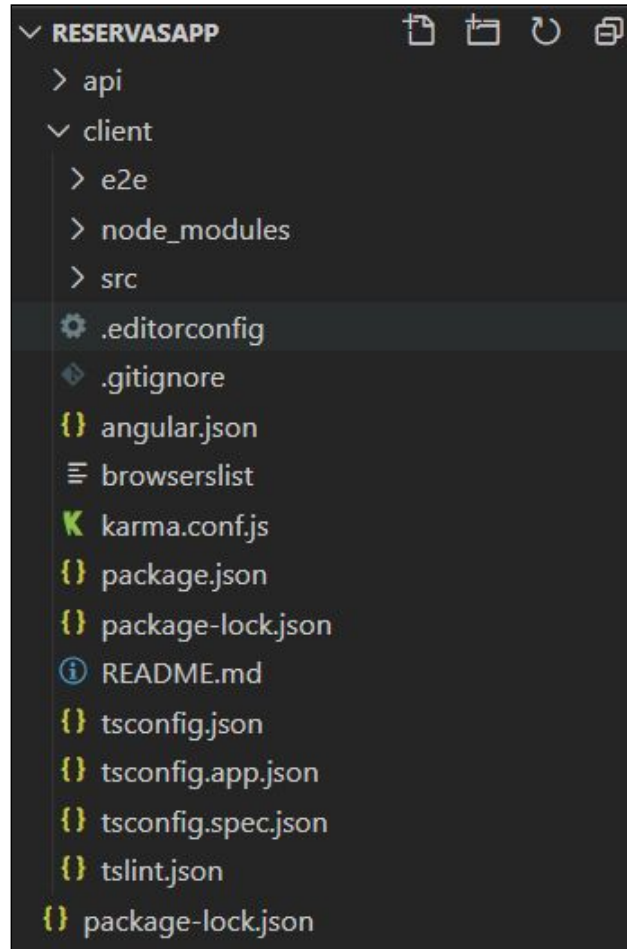


Figura 40: Proyecto en Angular

Dentro de src se encuentra el código de la aplicación, siendo index.html la página principal a la que hacemos referencia cuando hablamos de Single Page Application. También encontraremos los componentes que vamos creando y utilizando. El router se encargará de mostrar el componente deseado dependiendo del apartado de la web donde nos encontremos (Figura 41).

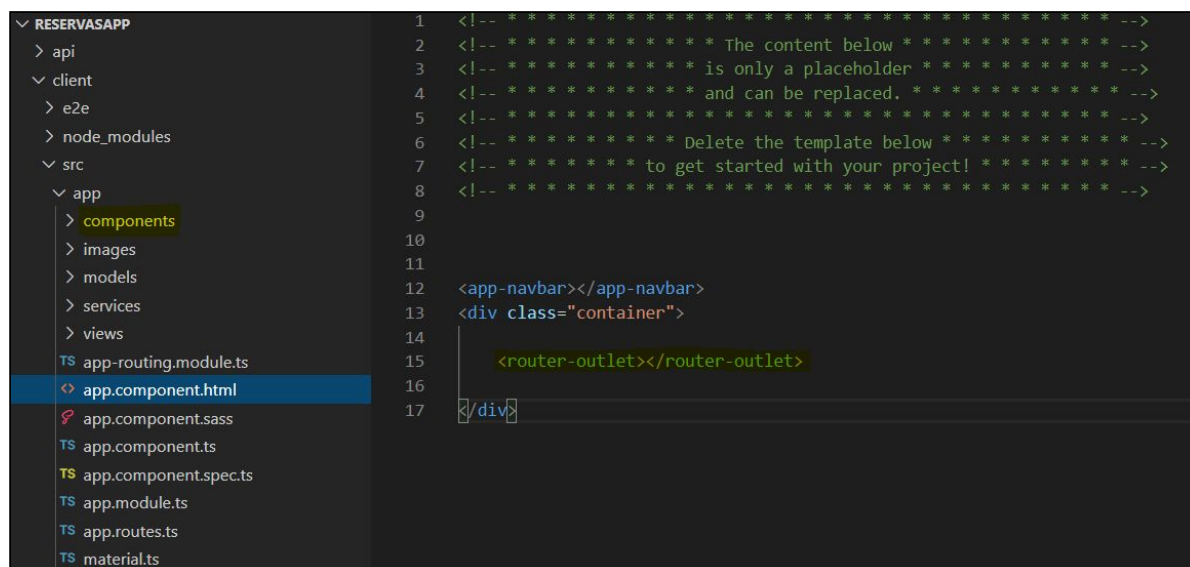


Figura 41: Single Page Application en Angular

Seguidamente, con el comando **ng serve**, situándonos en la raíz del proyecto, el sistema abrirá un servidor web con la aplicación web que hemos generado, y podremos visualizar los cambios conforme vamos escribiendo el código del mismo, normalmente a través del puerto 4200 (Figura 42).

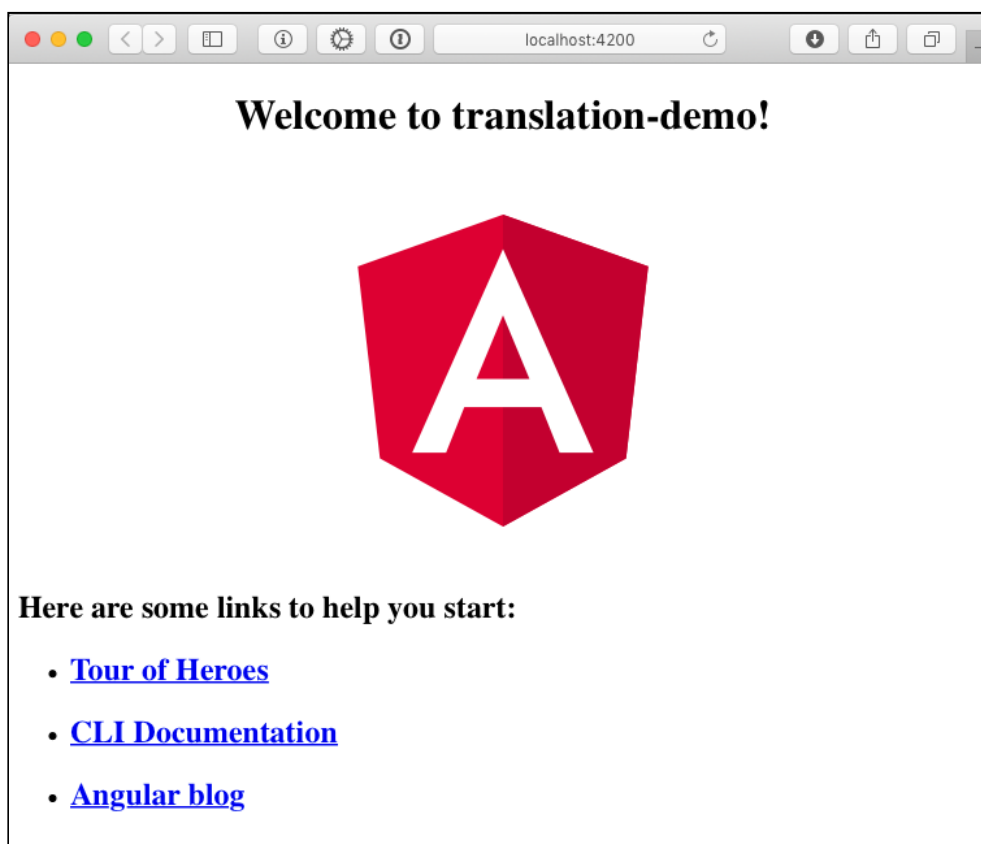


Figura 42: Servidor Web de la aplicación de prueba de Angular

Angular dispone, además, de una biblioteca de componentes llamada **Angular Material** (Figura 42) que podemos utilizar para nuestra página web y adaptarlos en la medida de lo necesario. Incluye formularios, botones, barras de navegación y prácticamente todo lo que vamos a necesitar para comenzar a programar (Figura 43).

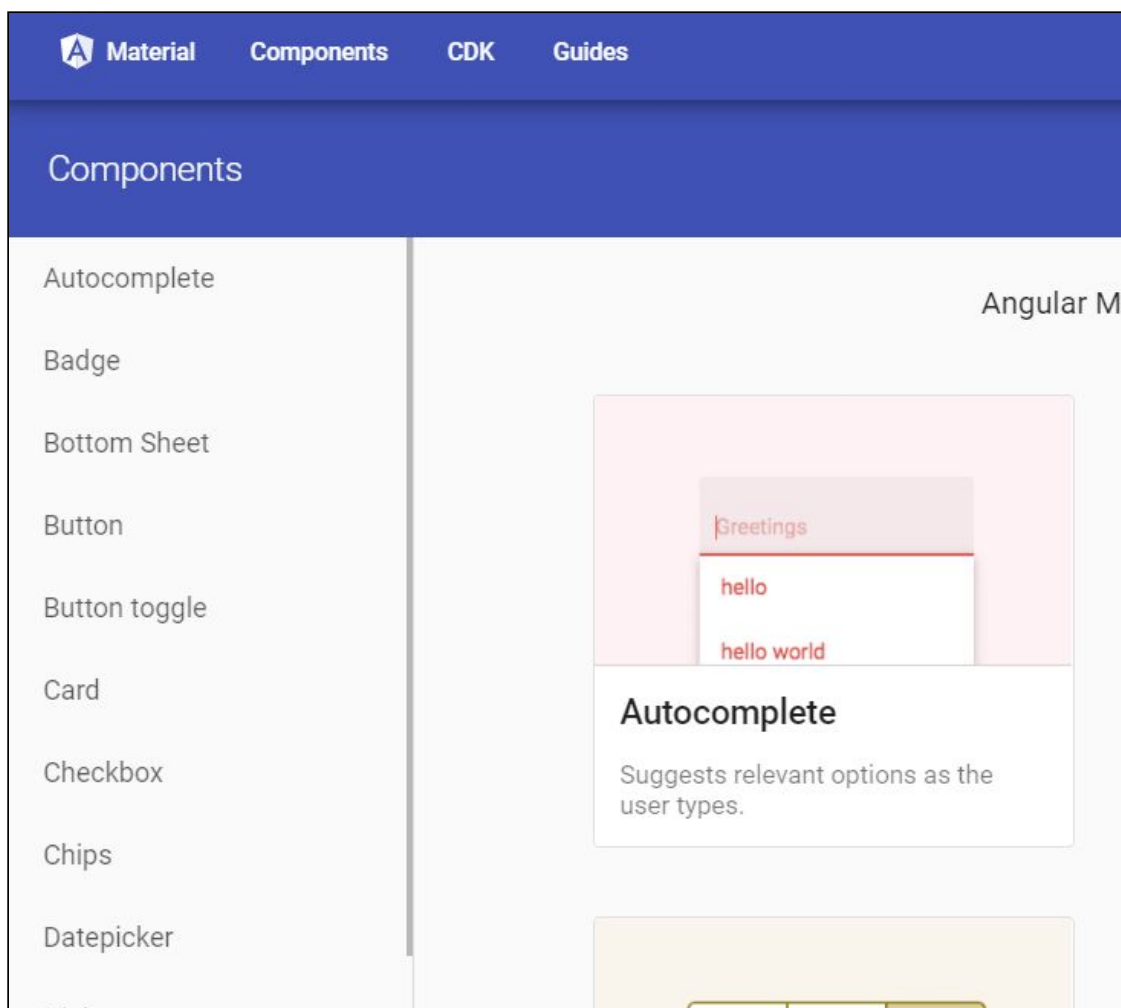


Figura 43: Buscador de extensiones de Visual Studio Code

Al generar el proyecto ya incluimos todos estos componentes en las dependencias, por lo que solo tenemos que ir a la documentación, buscar el nombre para importarlo y usar la etiqueta correspondiente en la parte HTML. En este caso, podemos ver la API para añadir un botón básico desde angular material a nuestro proyecto (Figura 44).



Figura 44: API de un botón de Angular Material

Y en el apartado Overview podemos ver todas las variantes de botones que incluye y sus etiquetas (Figura 45).

There are several button variants, each applied as an attribute:

Attribute	Description
mat-button	Rectangular text button w/ no elevation
mat-raised-button	Rectangular contained button w/ elevation
mat-flat-button	Rectangular contained button w/ no elevation
mat-stroked-button	Rectangular outlined button w/ no elevation
mat-icon-button	Circular button with a transparent background, meant to contain an icon
mat-fab	Circular button w/ elevation, defaults to theme's accent color
mat-mini-fab	Same as mat-fab but smaller

Figura 45: Variantes de Angular Button en la API de Angular Material

Además de todo esto, Angular es un conocido framework que tiene un portal de soporte bastante grande, por lo que podremos solucionar errores que vayan surgiendo sobre la marcha.

5.2.4 Node JS



Figura 46: Logo de Node JS

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Node.js brilla en aplicaciones web de tiempo real empleando la tecnología push a través de Websockets. Finalmente tenemos aplicaciones web en tiempo real, las conexiones bidireccionales, donde tanto el cliente como el servidor pueden iniciar la comunicación, lo que les permite intercambiar datos libremente. Esto está en contraste con el paradigma de respuesta web típica, donde el cliente siempre inicia la comunicación. Además, todo se basa en el Open Web Stack (HTML, CSS y JS) que se ejecuta en el puerto estándar 80.

Node.js no es lo más adecuado para operaciones intensivas de CPU; de hecho, utilizándolo para el cálculo pesado anulará casi todas sus ventajas. Donde Node realmente destaca es en la construcción rápida y escalable de aplicaciones de red, debido a que es capaz de manejar un gran número de conexiones simultáneas con alto rendimiento, lo que equivale a una alta escalabilidad.

5.2.4.1 NPM - Node Package Manager

Cuando hablamos de Node.js, una cosa que definitivamente no debe omitirse es integrarlo en el apoyo de la gestión de paquetes utilizando la herramienta NPM que viene por defecto con cada instalación de Node.js. Se trata de un conjunto de componentes reutilizables disponibles públicamente a través de una fácil instalación a través de un repositorio en línea, con la versión y la dependencia de gestión.

Una lista completa de los paquetes de módulos puede encontrarse en el sitio web de NPM (Figura 47) o acceder utilizando la herramienta de la CLI de NPM que automáticamente se instala con Node.js.

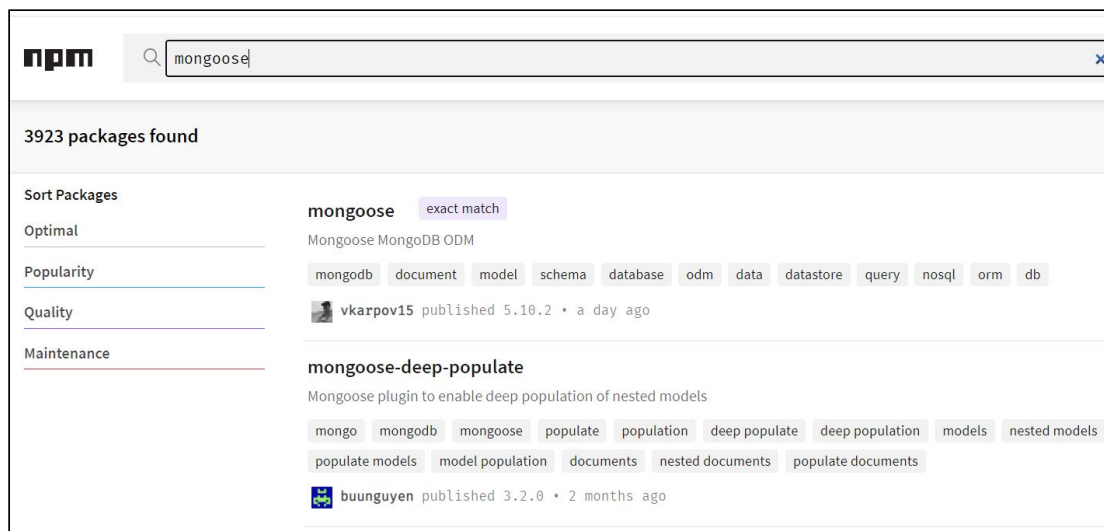


Figura 47: Buscador de extensiones de Node Package Manager

Para este proyecto, por ejemplo, utilizaremos la combinación de los componentes Mongoose y Express para conectar la aplicación a la base de datos NoSQL Mongo DB.

5.2.5 Mongoose + Express



Figura 48: Esquema para definir la unión de los paquetes Mongoose & Express JS

Como hemos comentado en el apartado anterior, Express y Mongoose son dos paquetes disponibles en el Node Package Manager que nos serán útiles para desarrollar esta aplicación.

Mongoose es una biblioteca de JavaScript que le permite definir esquemas con datos fuertemente tipados. Una vez que se define un esquema, Mongoose le permite crear un

Modelo basado en un esquema específico. Un modelo de mongoose se asigna a un documento MongoDB a través de la definición del esquema del modelo.

Una vez que haya definido sus esquemas y modelos, Mongoose contiene muchas funciones diferentes que le permiten validar, guardar, eliminar y consultar sus datos utilizando las funciones comunes de MongoDB.

Para la inicialización de mi aplicación, instalamos paquete de mongoose (Figura 49).

```
1 npm install mongoose --save
```

Figura 49: Comando de instalación de mongoose con Node Package Manager

Con todos los requisitos previos de la API configurados, nos conectamos a la base de datos de MongoDB colocando el comando necesario en index.js de la API (Figura 50).

```
1 var mongoose = require('mongoose');
2
3 mongoose.connect('mongodb://localhost/mongoose_basics');
```

Figura 50: Comando de conexión de la API al servidor de Mongo DB

Para hacer el código a prueba de fallos, podemos capturar errores que puedan surgir en la conexión (Figura 51).

```
var mongoose = require('mongoose');
var app = require('./app');
var port = process.env.PORT || 3977;

mongoose.Promise = global.Promise;
mongoose.connect('mongodb://localhost:27017/reservas', (err, res) =>{
  if(err){
    throw err;
  }else{
    console.log("La base de datos está corriendo correctamente...");
    app.listen(port, function(){
      console.log("Servidor del API Rest de reservas escuchando en http://localhost:"+port);
    });
  }
});
```

Figura 51: Código para conectar la API al servidor de Mongo DB

Express.js es un framework para Node.js que sirve para ayudarnos a crear aplicaciones web en menos tiempo ya que nos proporciona funcionalidades como el enrutamiento, opciones para gestionar sesiones y cookies, y un largo etc.

Express.js está basado en Connect, que a su vez es un framework basado en http para Node.js. Podemos decir que Connect tiene todas las opciones del módulo http que viene por defecto con Node y le suma funcionalidades. A su vez, Express hace lo mismo con Connect, con lo que tenemos un framework ligero, rápido y muy útil.

Primeramente, instalamos Express.js añadiéndolo a la lista de dependencias (Figura 52).

```
1
2 npm install express --save
3
```

Figura 52: Comando de instalación de Express JS con Node Package Manager

Seguidamente, dentro de index.js, copiamos algunos comandos de configuración (Figura 53).

```
1
2 var express = require('express');
3 var app = express();
4
```

Figura 53: Comandos de importación y ejecución de Express

A partir de aquí debemos seguir añadiendo algunas configuraciones como, por ejemplo, el código necesario para configurar las cabeceras HTTP (Figura 54).

```
// Configurar cabeceras http
app.use((req, res, next) => {
  res.header('Access-Control-Allow-Origin', '*');
  res.header('Access-Control-Allow-Headers', 'Authorization, X-API-KEY, Origin, X-Requested-With');
  res.header('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, DELETE');
  res.header('Allow', 'GET, POST, OPTIONS, PUT, DELETE');
  next();
})
```

Figura 54: Configuración de Cabeceras HTTP en index.js

6

Implementación y Pruebas

Una vez realizadas las configuraciones en Visual Studio Code, vamos a proceder a explicar la estructura del código de la aplicación. Primeramente empezaremos con la estructura de los archivos para entender cómo están organizados y, posteriormente, veremos qué componentes software Backend y Frontend se utilizan en los diferentes casos de uso y cómo los relacionamos.

6.1 API

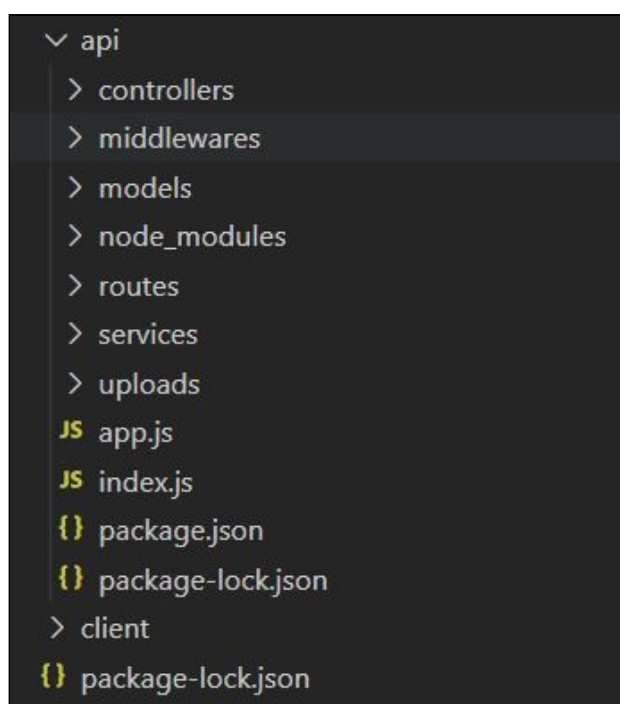


Figura 55: Directorio de la API del proyecto en Visual Studio Code

A continuación podemos ver la estructura de la API REST (Figura 55). Disponemos de una carpeta para los modelos, los controladores, rutas, servicios, middlewares y los archivos que subimos tales como imágenes, en la carpeta Uploads.

6.1.1 Modelos

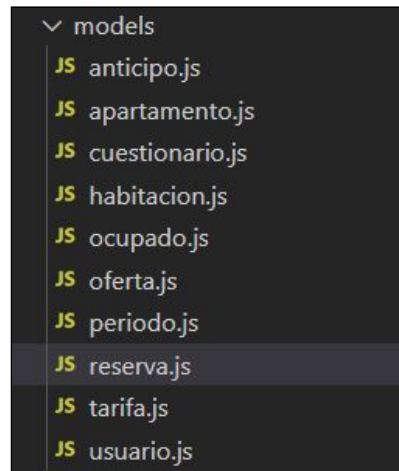


Figura 56: Directorio de Modelos de la API en Visual Studio Code

En los modelos definimos la estructura de las entidades previamente diseñadas en el Diagrama de Clases (Figura 56). Las tratamos como un objeto Java y exportamos como un modelo Mongoose.

```
api > models > JS reserva.js > ...
1  'use strict'
2
3  var mongoose = require('mongoose');
4  var Schema = mongoose.Schema;
5
6  var ReservaSchema = Schema({
7    f_inicio: Date,
8    f_fin: Date,
9    estado: String,
10   adultos: Number,
11   menores: Number,
12   precio: Number,
13   anticipo: Number,
14   valoracion: String,
15   usuario: {type: Schema.ObjectId, ref: 'Usuario'},
16   apartamentos: [{type: Schema.ObjectId, ref: 'Apartamento'}]
17 });
18
19 module.exports = mongoose.model('Reserva', ReservaSchema);
```

Figura 57: Ejemplo de Modelo de la API en Visual Studio Code

A continuación vemos como ejemplo el modelo de Reserva (Figura 57), con todos los campos y sus tipos. Definiendo estos modelos, podemos posteriormente definir posteriormente variables de este tipo para ejecutar peticiones a la base de datos, que trabajará con ellos y los recogerá en formato JSON.

6.1.2 Controladores

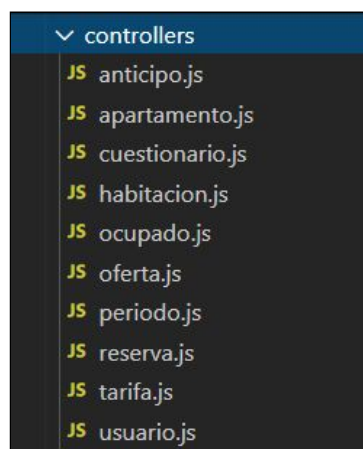


Figura 58: Directorio de Controladores de la API en Visual Studio Code

En los controladores importamos los modelos que hemos definido anteriormente e implementaremos los métodos que nos permitirán realizar peticiones a la base de datos (Figura 58).

```
3  var fs = require('fs');
4  var path = require('path');
5
6  var Habitacion = require('../models/habitacion');
7
8
9  > function saveRoom(req, res) { ...
30 }
31
32 > function getRooms(req, res) { ...
54 }
55
56 > function getRoom(req, res) { ...
70 }
71
72 > function deleteRoom(req, res) { ...
86 }
87
88 > function updateRoom(req, res) { ...
103 }
```

Figura 59: Controlador de Habitaciones en Visual Studio Code

Por ejemplo, en el caso de el controlador de habitaciones (Figura 59), definiremos las operaciones para guardar una reserva en la base datos, obtener las habitaciones de un apartamento, obtener una habitación mediante su ID o eliminar una habitación.

```
function getRooms(req, res) {
  var apartamentoId = req.params.idApartamento;

  var find = Habitacion.find({apartamento: apartamentoId});

  find.populate({
    path: 'apartamento',
    populate: {
      path: 'apartamento',
      model: 'Apartamento'
    }
  }).exec((err, habitaciones) =>{
    if(err){
      res.status(500).send({message: 'Error en la petición'});
    }else{
      if(!habitaciones){
        res.status(404).send({message: 'No hay habitaciones'});
      }else{
        res.status(200).send({habitaciones: habitaciones});
      }
    }
  });
}
```

Figura 60: Método para obtener las habitaciones de un apartamento en Visual Studio Code

A continuación podemos ver como ejemplo la función getRooms (Figura 60), que recoge la id del apartamento de los parámetros de la request y devuelve el conjunto de habitaciones, o bien un error si la petición no ha podido realizarse o no existen habitaciones para dicho apartamento.

6.1.3 Servicios



Figura 61: Directorio de Servicios de la API en Visual Studio Code

En la carpeta servicios tenemos el servicio JWT (Figura 61). Nos permitirá crear tokens de usuarios proporcionando un modelo usuario (Figura 62). Posteriormente, nos servirá para comprobar si se encuentran con la sesión iniciada y utilizar los datos.

```

var jwt = require('jwt-simple');
var moment = require('moment');
var secret = 'superreservado';

exports.createToken = function (usuario) {
  var payload = {
    sub: usuario._id,
    dni_pass: usuario.dni_pass,
    nombre: usuario.nombre,
    apellidos: usuario.apellidos,
    email: usuario.email,
    rol: usuario.rol,
    f_nac: usuario.f_nac,
    telefono: usuario.telefono,
    iat: moment().unix(),
    exp: moment().add(30, 'days').unix()
  }

  return jwt.encode(payload, secret);
}

```

Figura 62: Método de creación de Token en Visual Studio Code

JWT se trata de una pequeña biblioteca de navegador que ayuda a decodificar tokens que están codificados en Base64Url.

6.1.4 Middlewares

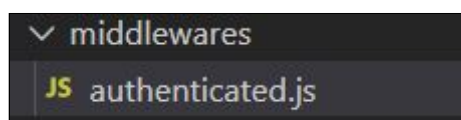


Figura 63: Directorio de Middlewares de la API en Visual Studio Code

El middleware authenticated (Figura 63) nos permitirá comprobar si un token es válido antes de realizar un servicio.

```

var jwt = require('jwt-simple');
var moment = require('moment');
var secret = 'superreservado';

exports.ensureAuth = function(req, res, next){
  if(!req.headers.authorization){
    return res.status(403).send({message: "La petición no contiene la cabecera de autorización"});
  }else{
    var token = req.headers.authorization.replace(/["]+/g, '');
    try{
      var payload = jwt.decode(token, secret);
      if(payload.exp <= moment.unix()){
        return res.status(401).send({message: "El token ha expirado"});
      }
    }catch(ex){
      console.log(ex);
      return res.status(404).send({message: "Token no válido"});
    }
    req.usuario = payload;
    next();
  }
}

```

Figura 64: Método de comprobación del Token en Visual Studio Code

Primeramente se recoge el token de la cabecera de autenticación de la petición y, en caso de que haya, comprueba si la fecha de expiración `payload.exp` es menor a este momento actual, utilizando `moment` para capturar el momento exacto de tiempo en el que nos encontramos, lo cual significaría que el token ha caducado (Figura 64).

6.1.5 Rutas

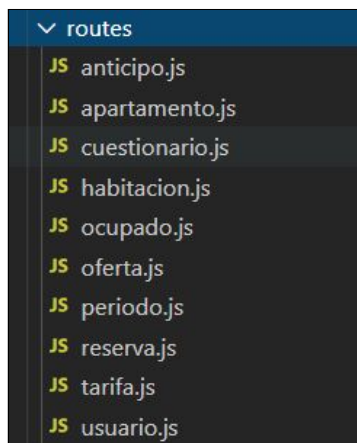


Figura 65: Directorio de Rutas de la API en Visual Studio Code

Las rutas (Figura 65) nos permitirán definir como realizaremos las peticiones a la base de datos, introduciendo la ruta y el método del controlador al que redirigirá.

```
var express = require('express')
var ControladorDePeriodo = require('../controllers/periodo');
var md_auth = require('../middlewares/authenticated')

var api = express.Router();

api.post('/periodo/', md_auth.ensureAuth, ControladorDePeriodo.savePeriod);
api.get('/periodos/:tarifa', ControladorDePeriodo.getPeriodsByTariff);
api.delete('/periodo/:id', md_auth.ensureAuth, ControladorDePeriodo.deletePeriod);
api.put('/periodo/:id', md_auth.ensureAuth, ControladorDePeriodo.updatePeriod);

module.exports = api;
```

Figura 66: Archivo de Rutas para la tabla Periodos de la API en Visual Studio Code

En el caso de las rutas de periodos, por ejemplo, definimos las peticiones HTTP (PUT, GET, DELETE ó UPDATE) para crear un periodo, obtener los periodos de una tarifa determinada, eliminar un periodo o modificarlo indicando su id (Figura 66).

Añadimos el signo ':' detrás de la ruta para indicar que se trata de un parámetro necesario para ejecutar el método, y ejecutamos el método 'ensureAuth' en aquellos métodos en los que será necesario que el usuario esté conectado. Por ejemplo, para eliminar un periodo, es estrictamente necesario que el usuario esté conectado, pero no hace falta para obtener los periodos de una tarifa.

6.2 FRONT

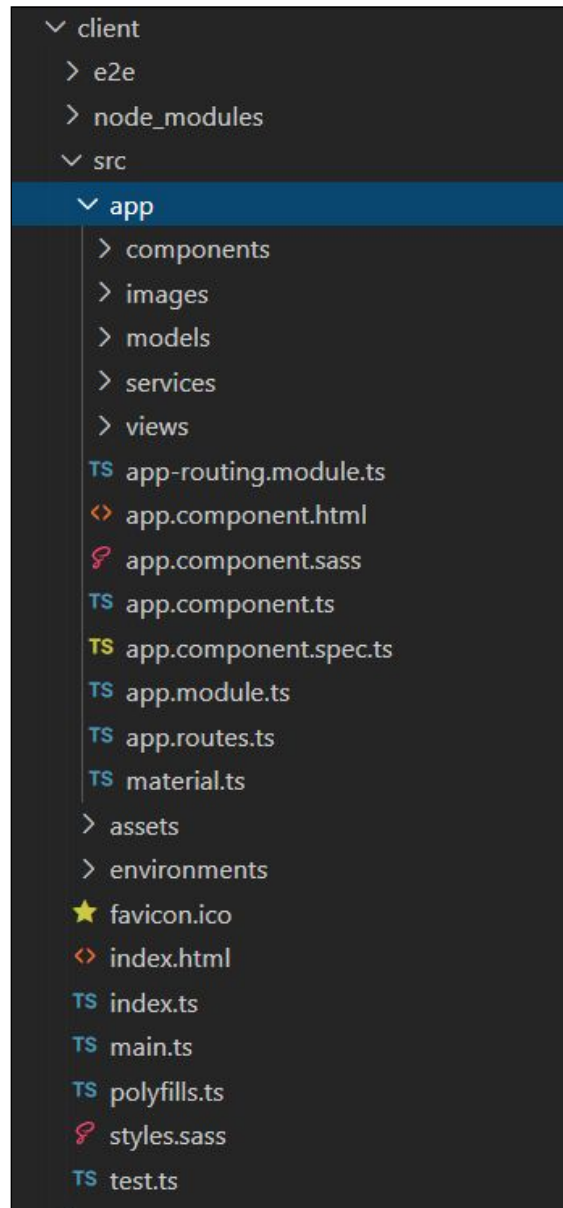


Figura 67: Directorio del Proyecto Front de Angular en Visual Studio Code

A continuación podemos ver la estructura de la parte front de la aplicación (Figura 67). Como sabemos, se trata de una Single Page Application. Por un lado tenemos la página front en 'index.html' e 'index.ts', las configuraciones correspondientes y por otra parte una carpeta para los componentes, sus vistas, servicios y modelos.

6.2.1 Modelos

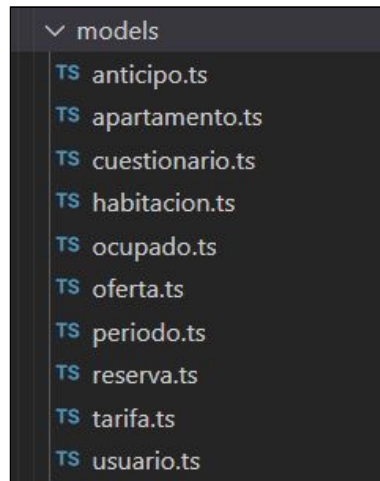


Figura 68: Directorio de Modelos para la parte Front en Visual Studio Code

De la misma forma que para la parte Backend, debemos definir los modelos que usaremos en los componentes de la web (Figura 68). La diferencia es que en este caso, cuando tenemos una clave foránea a otro modelo lo definimos como un String, ya que guardaremos la ID.

```
export class Reserva{  
  constructor(  
    public _id: string,  
    public f_inicio: Date,  
    public f_fin: Date,  
    public estado: String,  
    public adultos: Number,  
    public menores: Number,  
    public precio: number,  
    public anticipo: Number,  
    public usuario: String,  
    public valoracion: Number,  
    public apartamentos: String[]  
  ){}  
}
```

Figura 69: Modelo de Reserva para la parte Front en Visual Studio Code

En el ejemplo de una Reserva, los campos necesarios y el tipado correspondiente, añadiendo también el campo ‘_id’ que se genera automáticamente cuando se crea un nuevo objeto en Mongo DB (Figura 69).

6.2.2 Servicios

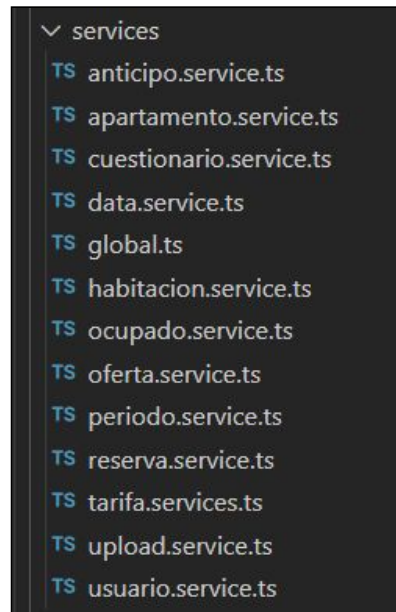


Figura 70: Directorio de Servicios para la parte Front en Visual Studio Code

De la misma forma que en el backend, en la parte front también creamos servicios para cada entidad (Figura 70), que nos permitirán realizar las peticiones HTTP y llamar a los métodos de los controladores para trabajar con la base de datos.

```
@Injectable()
export class HabitacionService {
  public url: string;

  constructor(private _http: Http) {
    this.url = GLOBAL.url;
  }

  > addRoom(token, habitacion: Habitacion){ ...
  }

  > getRoom(id: string){ ...
  }

  > updateRoom(token, id: string, habitacion: Habitacion){ ...
  }

  > getRooms(idApartamento: string){ ...
  }

  > deleteRoom(token, id: string){ ...
  }
```

Figura 71: Servicio para el modelo de Habitación de la parte Front en Visual Studio Code

Para el servicio del modelo de habitación (Figura 71), por ejemplo, disponemos de un método para añadir, obtener, modificar y eliminar habitaciones y un método para obtener todas las habitaciones de un apartamento concreto mediante su ID.

```
updateRoom(token, id: string, habitacion: Habitacion){  
  let params = JSON.stringify(habitacion);  
  let headers = new Headers({  
    'Content-Type': 'application/json',  
    'Authorization': token  
  });  
  
  return this._http.put(this.url+'habitacion/'+id, params, {headers: headers}).map(res => res.json());  
}
```

Figura 72: Método del servicio de Habitación de la parte Front para actualizar una habitación en Visual Studio Code

En el caso del método ‘updateRoom’ del servicio de habitación, por ejemplo, que nos permitirá modificar una habitación con una ID determinada, pasamos por parámetro la ID de la habitación, el token del usuario y el objeto habitación cuyo modelo hemos creado previamente (Figura 72).

Generamos la petición HTTP de tipo PUT y añadimos la ruta que definimos en la API, añadiendo la ID tras la ruta y la cabecera de autorización con el token de usuario.

6.2.3 Single Page Application

Como ya sabemos, Angular se utiliza para crear Single Page Applications. Esto quiere decir que toda la aplicación se implementa en una sola página, la cual va cargando componentes a través de un router para navegar por la misma.

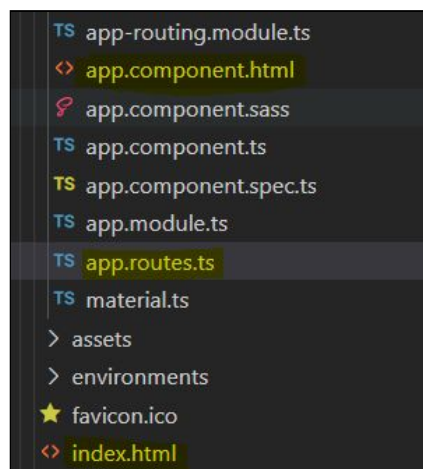


Figura 73: Localización de la Single Page Application y el Router en el proyecto de Angular

La página principal es ‘index.html’ (Figura 74), que es la que abre el servidor web por defecto.

```

<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Reservasapp</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">

  <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500&display=swap" rel="stylesheet">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
</head>

<body>
  <app-root></app-root>
</body>

</html>

```

Figura 74: Página principal del proyecto Front en Visual Studio Code

Como vemos, lo único que incluye el cuerpo de la web es el ‘app-root’, que se encarga de cargar el componente principal ‘app-component’ (Figura 75).

```

<app-navbar></app-navbar>
<div class="container">
  <router-outlet></router-outlet>
</div>

```

Figura 75: Componente principal del proyecto Front en Visual Studio Code

Como vemos, app-component solo contiene el componente de la barra de navegación ‘app-navbar’ y el ‘router-outlet’, que se encarga de cargar cada componente.

```

const APP_ROUTES: Routes = [
  { path: 'home', component: HomeComponent },
  { path: 'apartamentos', component: ApartamentosComponent },
  { path: 'misreservas/:id', component: MisReservasComponent },
  { path: 'apartamento/:id', component: ApartamentoPageComponent },
  { path: 'cuestionario/:id', component: CuestionarioComponent },
  { path: 'conectarse', component: ConectarseComponent },
  { path: 'reservar/:entrada/:salida', component: ReservarComponent },
  { path: 'panel-admin', component: PanelAdminComponent },
  { path: '**', pathMatch: 'full', redirectTo: 'home' }
];

```

Figura 76: Configuración de rutas y componentes en el proyecto Front en Visual Studio Code

Esta configuración se encuentra dentro de app-route.ts, que nos permite configurar qué componente vamos a cargar según la ruta en la que nos encontremos (Figura 76). Por defecto, si no introducimos ninguna ruta, nos redirigirá por defecto al componente ‘home’ indicando como ruta ‘**’, es decir, cualquiera que no sea ninguna de las anteriores.

6.2.4 Componentes

Como ya sabemos, el router se encarga de cargar cada componente, el cual se compone de su archivo typescript y su vista html.

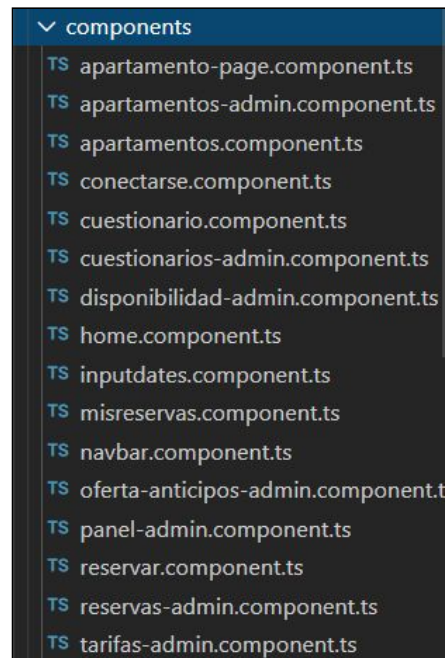


Figura 77: Directorio de Componentes del proyecto de Angular en Visual Studio Code

Tenemos un archivo typescript de configuración para cada componente de la web (Figura 77). Dentro de él encontramos todo lo necesario para cargar la vista correctamente.

```
public today: Date;
public url: string;
public reservas: [Reserva];
public f_inicio: Date;
public f_fin: Date;
public mapQuest = new Map<string, boolean>();
public identity;
public token;
public errorMessage: String;

constructor(
  private _userService: UsuarioService,
  private _bookingService: ReservaService,
  private _questionaryService: CuestionarioService,
  private _route: ActivatedRoute
) {
  this.errorMessage = null;
  this.today = moment().toDate();
  this.url = GLOBAL.url;
  this.f_inicio = null;
  this.f_fin = null;
}
```

Figura 78: Archivo TypeScript de configuración de un componente del proyecto Angular en Visual Studio Code

Dentro de cada componente podemos declarar variables necesarias que serán leídas en la vista HTML, así como llamar a servicios para hacer peticiones a la API. Por ejemplo, en el caso de la página ‘Mis Reservas’, necesitaremos el servicio de usuario para recuperar el token, el de reservas para recuperar las reservas del usuario y el servicio de cuestionarios para obtener los cuestionarios de cada reserva correspondiente o rellenar uno nuevo (Figura 78).



```
getBookings() {
  this._route.params.forEach((params: Params) => {
    let id = params['id'];
    this._bookingService.getBookingsByID(this.token, id).subscribe(
      (response) => {
        if (response.reservas.length > 0) {
          for(let reserva of response.reservas){
            this.setQuestionaryByBookingID(reserva._id);
          }
          this.reservas = response.reservas;
        }else{
          this.errorMessage = "Todavía no has hecho ninguna reserva, cuando la realices podrás ver los detalles en este apartado."
        }
      },
      err => {
        var errorMessage = <any>err;
        if (errorMessage != null) { ...
      }
    }
  })
}
```

Figura 79: Método para obtener las reservas de un usuario suministrando su identificador y un token válido

Por ejemplo, aquí podemos ver como desde el componente llamamos al servicio de reservas para obtener todas las reservas de un usuario dada una ID que tenemos como parámetro en la ruta (Figura 79).

6.2.5 Vistas

Por último, las vistas son los archivos HTML de los componentes, que se encargan de pintar las pantallas usando los datos que recogemos de las variables de sus archivos typescript correspondientes (Figura 80).

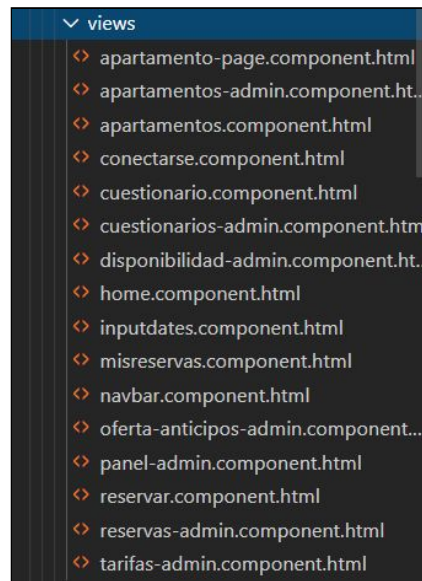


Figura 80: Directorio de las vistas de los componentes en el proyecto de Angular en Visual Studio Code

Vamos a ver un ejemplo directo en relación al archivo typescript del componente ‘Mis Reservas’ que hemos visto en el apartado anterior.

```
<div *ngFor="let reserva of reservas" class="card"
  style="width: 100%; flex-flow:inherit; margin-bottom: 10px; margin-top: 30px;">
  <div class="card-body">
    <h5 style="display:flex;" class="card-title"> Apartamentos:
      <div style="justify-content:center; padding-left: 5px;" *ngFor="let apartamento of reserva.apartamentos">
        <div *ngIf="apartamento == reserva.apartamentos[reserva.apartamentos.length-1]">
          {{apartamento.nombre}}
        </div>
        <div *ngIf="apartamento != reserva.apartamentos[reserva.apartamentos.length-1]">
          {{apartamento.nombre}},
        </div>
      </div>
    </div>
  </h5>
```

Figura 81: Fragmento de código de la vista del componente ‘Mis Reservas’ del proyecto de Angular en Visual Studio Code

En este pequeño fragmento de código (Figura 81) podemos ver como, mediante un bucle for, pintamos todas las reservas que recogimos en la variable ‘reservas’ llamando al servicio de reservas. Para cada iteración, pintamos una tarjeta y el nombre de los apartamentos que intervienen en la reserva.

6.3 Pruebas

Una vez hemos implementado toda la aplicación debemos realizar pruebas para comprobar que la página funciona correctamente. Para ellos haremos pruebas unitarias en la base de datos utilizando Postman y pruebas de integración utilizando Mongo Compass.

6.3.1 Pruebas Unitarias



Figura 82: Logo de Postman

Postman nace como una herramienta que principalmente nos permite crear peticiones sobre APIs de una forma muy sencilla y poder, de esta manera, probar las APIs. El usuario de Postman puede ser un desarrollador que esté comprobando el funcionamiento de una API para desarrollar sobre ella o un operador el cual esté realizando tareas de monitorización sobre un API.

De esta manera, permitirá realizar peticiones mientras se desarrolla la API Rest de la aplicación y comprobar que los métodos funcionan correctamente.

Por ejemplo, vamos a realizar una petición para recuperar un objeto de tipo ocupado, añadiendo en la URL, como definimos en las rutas, la ID del apartamento y la fecha como parámetros.

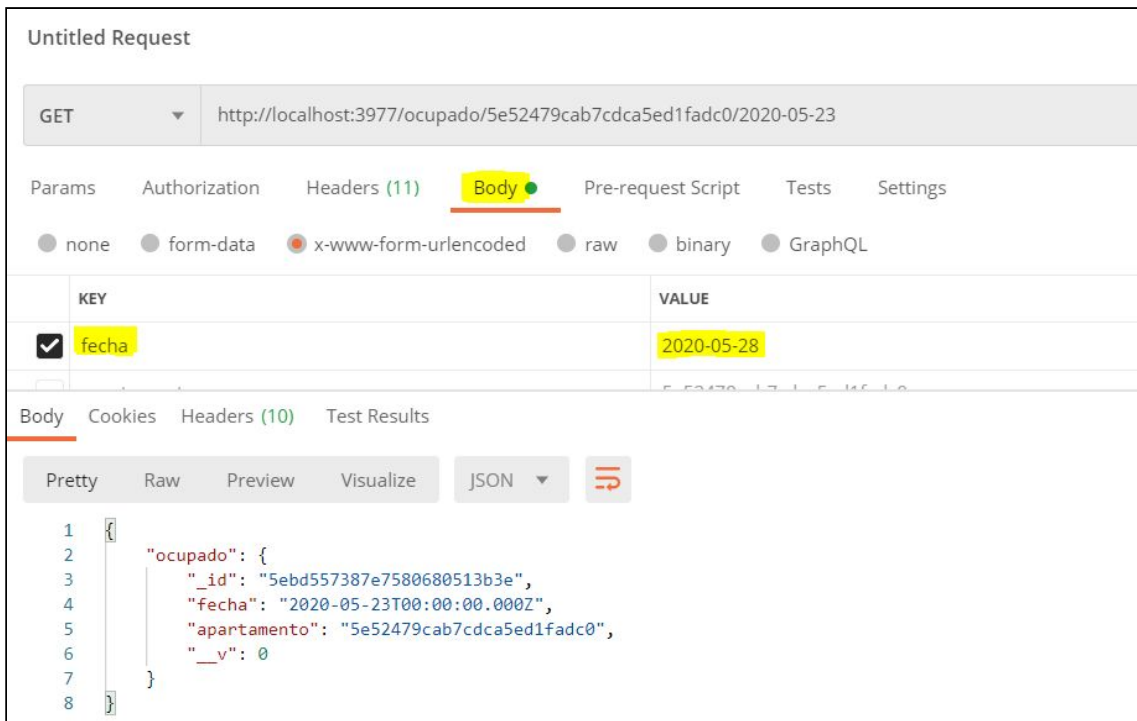


Figura 83: Consulta de un objeto de la clase Ocupado devuelto en formato JSON desde Postman

Como podemos ver, ya que existe un objeto de tipo ocupado asociado a dicho apartamento y con la fecha introducida, el sistema nos devuelve el objeto en formato JSON (Figura 83). Podemos realizar peticiones GET, POST, UPDATE y DELETE y más. Para asegurarnos de que la web funcione correctamente, debemos cerciorarnos de que funcionan todos los métodos de todas las rutas indicadas en las rutas de la API.

Podemos añadir los campos necesarios en el Body cuando sean necesarios y la cabecera de autenticación en Header (Figura 84), si esa petición requiere estar conectado añadiendo el token.

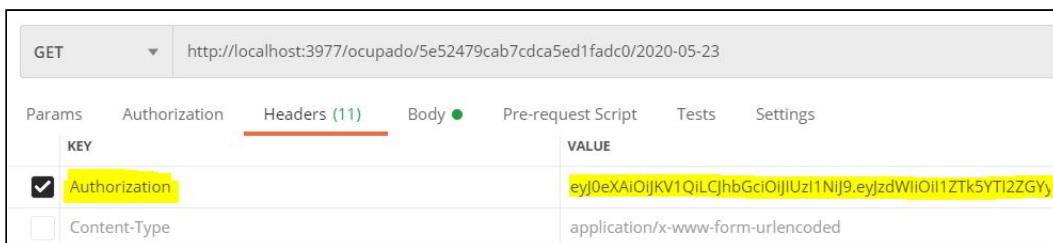


Figura 84: Cabecera de Autorización para el Token desde Postman

Si algo no es correcto, el sistema nos lanza el error correspondiente, por ejemplo si el token ha expirado o nos hemos equivocado escribiendo la petición (Figura 85).

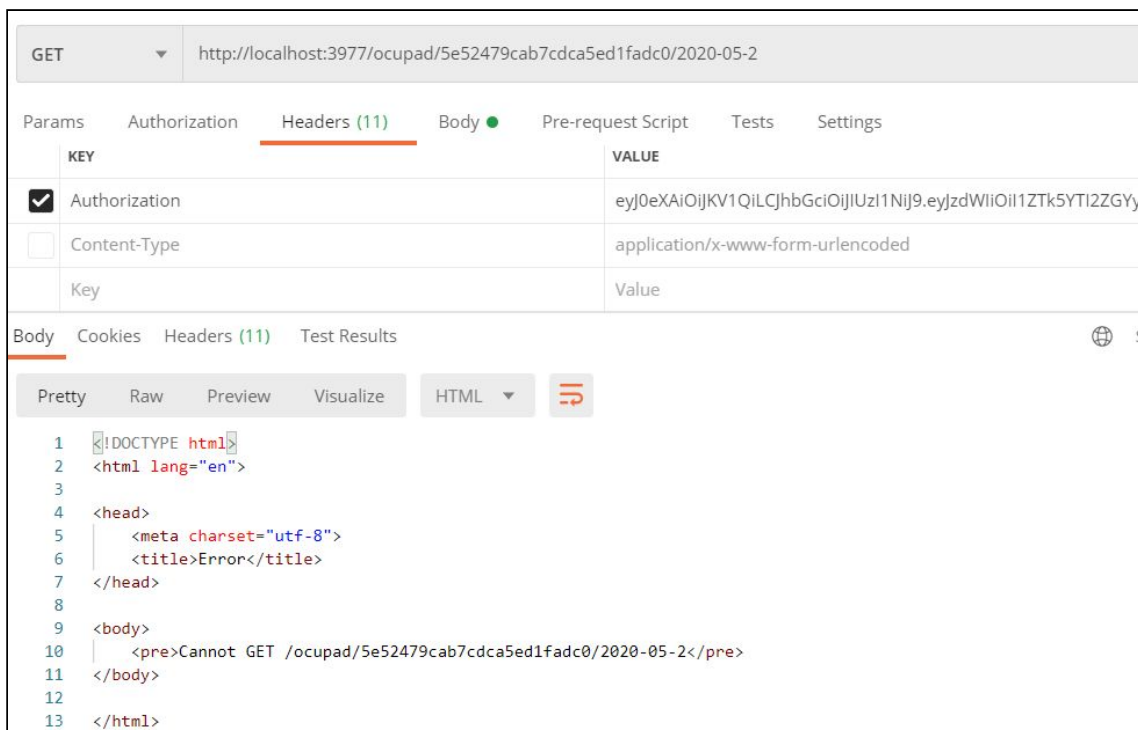


Figura 85: Ejemplo de un error en una consulta desde Postman por error en la URL

Por ejemplo, podemos ver un ejemplo en el cual tratamos de realizar una petición que requiere una cabecera de autorización con Token sin incluirla, donde el sistema nos lanzará un error, como por ejemplo el método para obtener los cuestionarios de todas las reservas (Figura 86).

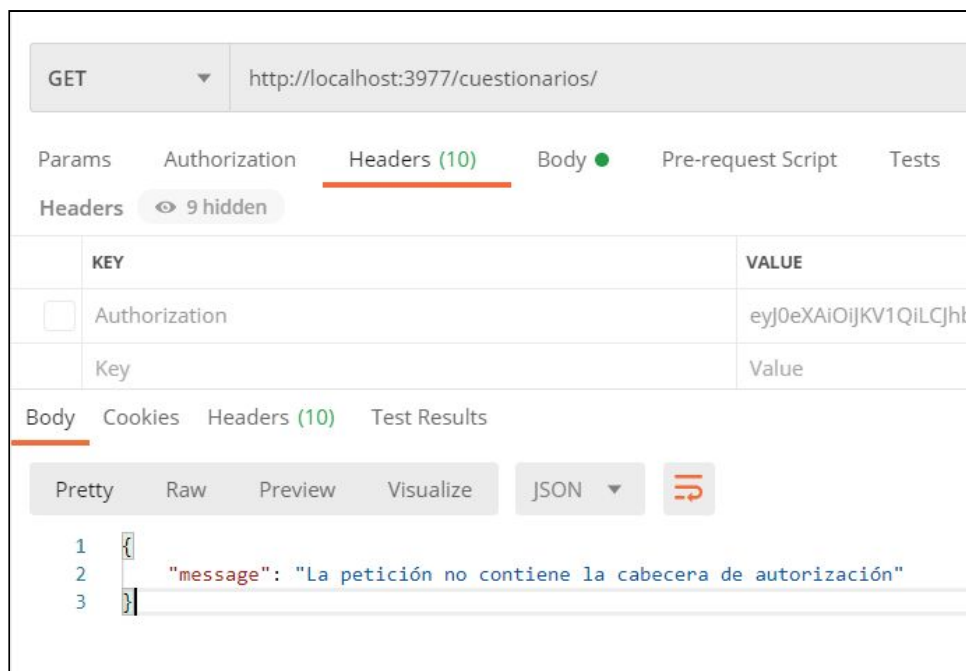


Figura 86: Ejemplo de error en la petición por Token no válido desde Postman

6.3.2 Pruebas de Integración

Una vez hemos realizado todas las pruebas unitarias necesarias, debemos comprobar que todo funciona correctamente y que el FRONT se comunica con la API para realizar las peticiones de manera correcta.

Para ello, utilizaremos la misma aplicación ya desarrollada y Mongo Compass.

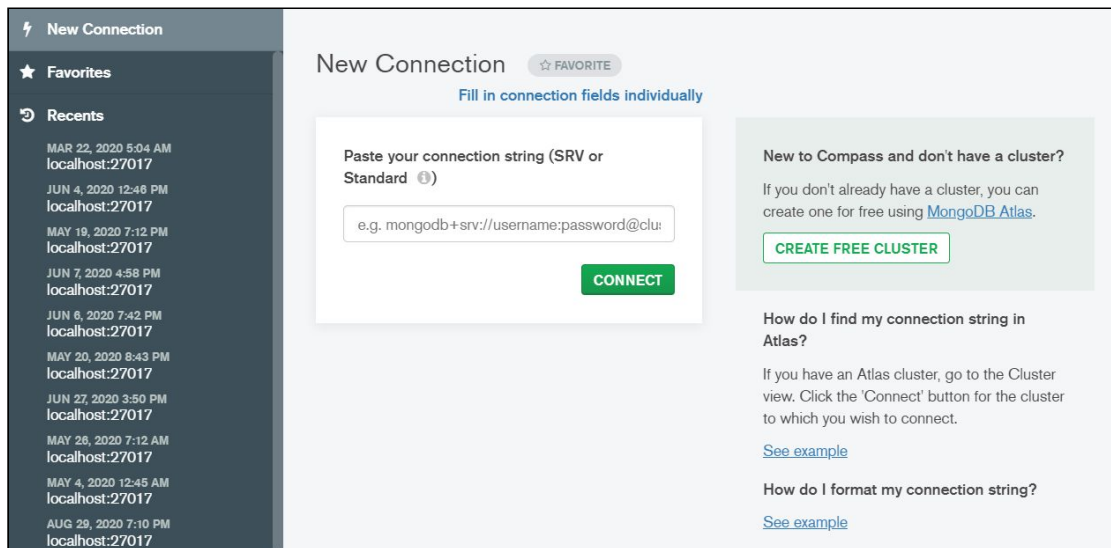


Figura 87: Vista principal para conectarse a un servidor desde Mongo Compass

Para conectarnos a la base de datos de forma visual, abrimos Mongo Compass e introducimos el servidor en el que estamos corriendo la base de datos (Figura 87).

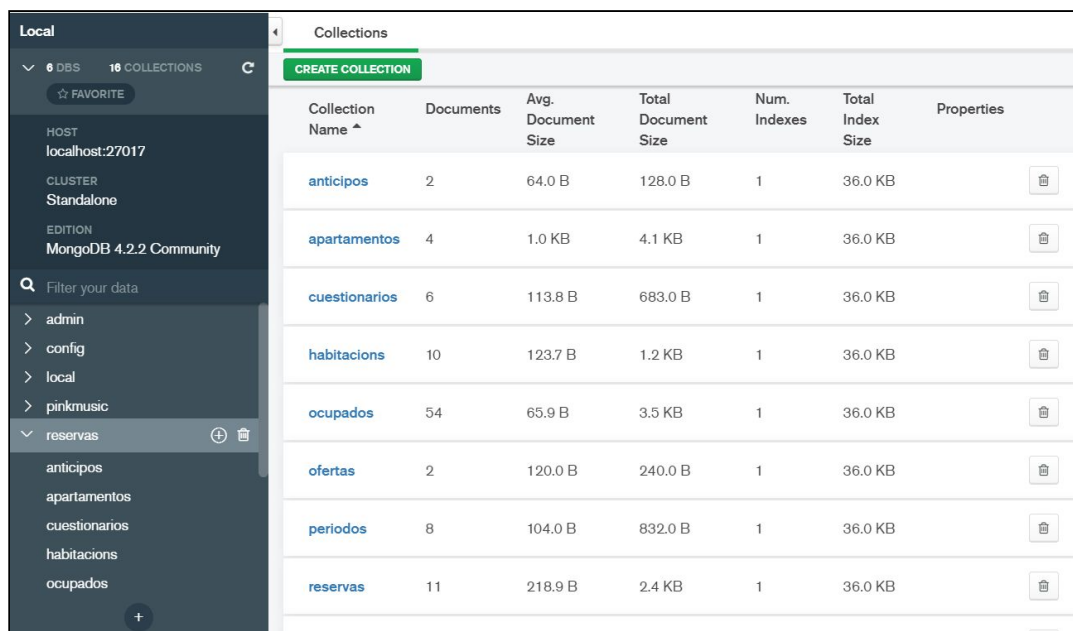


Figura 88: Vista de las colecciones de la base de datos 'reservas' desde Mongo Compass

Una vez que hemos accedido tenemos acceso a las bases de datos, y si presionamos en nuestra base de datos, tendremos acceso a todas las colecciones (Figura 88).

6.3.2.1 Proceso de Comprobación

Para comprobar el correcto funcionamiento, lo que haremos es utilizar la misma aplicación ya desarrollada y manejar las distintas funcionalidades. Por ejemplo, vamos a crear un nuevo apartamento y después nos aseguraremos de que dicho apartamento se ha creado en la base de datos.



Figura 89: Vista de creación de un nuevo apartamento desde el servidor web

Nos situamos en el Panel de Administrador, conectados como usuario Administrador y a punto de crear un nuevo apartamento. Introducimos los datos correctamente y pulsamos en el botón ‘Crear Apartamento’ (Figura 89) .

Figura 90: Vista de edición de un apartamento desde el servidor web

El sistema nos envía a la página de edición del apartamento (Figura 90), tal y como estaba programado en los diagramas de secuencia. No obstante, ahora iremos a Mongo Compass para comprobar que dicho apartamento ha sido creado con los datos correctos.

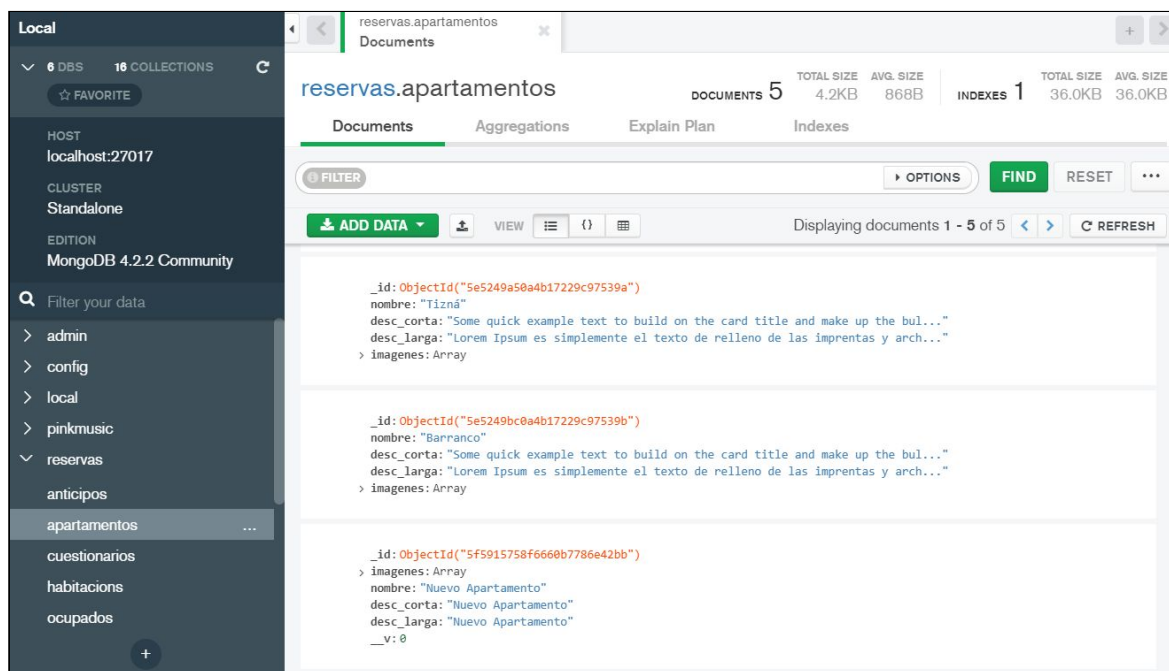


Figura 91: Vista de la colección de reservas desde Mongo Compass

Como podemos comprobar, el nuevo objeto de tipo apartamento se ha creado con los datos que hemos introducido (Figura 91).

Además, para mayor facilidad, Mongo dispone de un filtro en el que se pueden realizar consultas para visualizar y encontrar los datos que deseamos más fácilmente. Por ejemplo,

vamos a indicar que busque los apartamentos cuyo atributo 'nombre' sea "Nuevo Apartamento" (Figura 92).

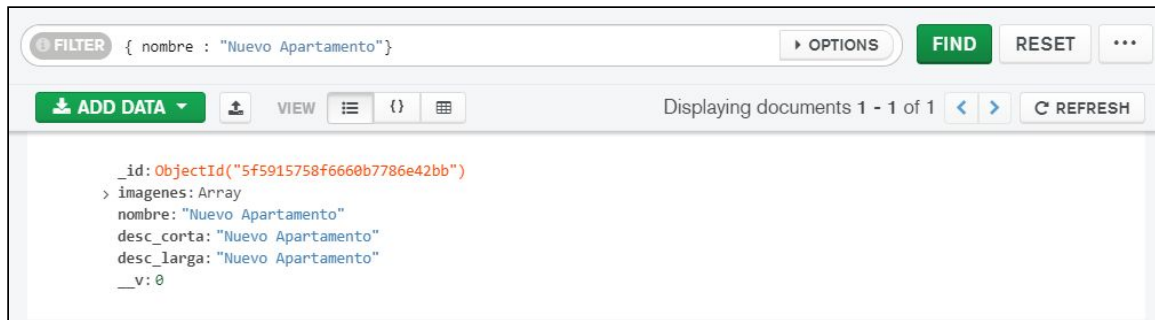


Figura 92: Consulta de un apartamento por campo nombre desde Mongo Compass

Este proceso debemos repetirlo para probar todas las funcionalidades de la web, asegurándonos de esta manera de que todo funciona perfectamente y está correctamente integrado.

Conclusiones

En este capítulo vamos a redactar cuáles han sido mis sensaciones tras haber finalizado el proyecto. Además se detallarán algunas mejoras con vistas a futuro para la aplicación, sobre todo tareas que han surgido sobre la marcha y han quedado pendientes de implementación por no haber entrado en la planificación.

Haber finalizado este trabajo ha supuesto un gran aprendizaje para mí. En primer lugar se trata del primer trabajo que he realizado de forma independiente, no en grupo, haciendo cada una de las partes del proyecto. He ampliado conocimientos con respecto a lo cursado en la carrera, lo cual me ha permitido conocer un poco tecnologías más punteras, como son las bases de datos no relacionales como Mongo DB o un potente framework para desarrollo del front-end como Angular.

Me he dado cuenta de que la planificación es muy importante, sobre todo en proyectos informáticos. Cuando planificas, te das cuenta de errores que empezando a programar directamente quizás se pasan por alto. Ya no solo porque se pasen por alto a ti como programador, sino porque en ocasiones el cliente puede no saber lo que quiere desde un principio, y al ver la maqueta, por ejemplo, puede detectar que algo que pensaba que quería en realidad lo quiere de otra manera. De hecho, las funcionalidades de este proyecto cambiaron desde los objetivos de la introducción hasta el desarrollo que finalmente se implementó. Además, el estudio de mercado te permite comprobar si lo que vas a programar ya existe en el mercado, lo cual te permite saber si lo que vas a hacer es innovador y plantearte si realmente merece la pena seguir adelante.

Por otra parte, cuando se definen todos los diagramas casi salen todos los métodos a implementar a la hora de programar, por lo que pienso que se puede estimar la duración de un proyecto. Además, definirlo de una manera tan exacta me hace pensar que cualquier

programador podría ver la planificación y entender la aplicación a la perfección sin necesidad de una persona que se lo explique.

De cara al futuro, pienso que la web podría mejorarse en cuanto aspecto y diseño. Además, antes de subirla a internet se debería de modificar para que sea responsive e implementar algunas funcionalidades restantes como un mailing para notificaciones a clientes, opción de he olvidado mi contraseña y gestor de usuarios para el administrador.

Mi idea hacia el futuro es desarrollarme como Full Stack Developer, por lo que esto ha sido una primera toma de contacto en la que he tocado un poco de cada cosa. Desde luego, pienso que si decido seguir con mis estudios y realizar un Master voy a llevar ventaja, ya que he ampliado conocimientos y he tocado algunos ámbitos de la programación que hasta ahora no me había decidido a experimentar por miedo a salir de mi zona de confort.

En general, estoy bastante contento con el trabajo realizado. No solo ha supuesto una formación para mi en el ámbito profesional sino una mejora en la calidad de vida de mi famili

Referencias

- Booch, G., Rumbaugh, J. y Jacobson, I. *El Lenguaje Unificado de Modelado*. Addison Wesley, 2009
- Pressman, R. *Ingeniería del Software*. McGraw Hill, 2009
- Victor Robles. *Desarrollo web con JavaScript, Angular, NodeJS y MongoDB*. Udemy, 2020.
- Luis Fernández Muñoz. *Análisis y Diseño Orientados a Objetos*. Desarrolloweb, 2015: <https://cutt.ly/LfVBBly>
- Tyrone. *Magicdraw: ¿Qué es MagicDraw? ¿Para qué sirve?* Youtube, 2015: <https://cutt.ly/mfVNRXy>
- Ruben. *MongoDB: Qué es, cómo funciona y cuándo podemos usarlo*. Genbeta, 2014: <https://cutt.ly/bfVB1il>
- Ruben. *MongoDB: Encuentra tus datos con consultas simples*. Genbeta 2014: <https://cutt.ly/jfVNx2k>
- Cristina Reyes Miguel. *Qué es un análisis DAFO: usos y ejemplos*. IEBSchool, 2019: <https://cutt.ly/NfVV8cI>
- Julio Roche. *Scrum: roles y responsabilidades*. Deloitte: <https://cutt.ly/lfVBF5U>
- Angular. *Components*. Angular Material, 2020: <https://material.angular.io/components/categories>
- Bootstrap. *Components*. Bootstrap Documentation, 2020: <https://getbootstrap.com/2.3.2/components.html>

Apéndice A

Manual de Usuario

En este apéndice vamos a elaborar un manual de usuario en función de los roles con los que se puede usar la aplicación, utilizando todas las funcionalidades de la misma.

A.1 Usuario no identificado

Cuando entramos a la web como usuario no identificado tenemos acceso a apartados de la web; Inicio & Apartamentos (Figura 93). Dentro de estas secciones podemos realizar varias funcionalidades; consultar la oferta actual, realizar una reserva, conectarnos como usuario, visualizar el calendario de disponibilidad de apartamentos y visualizar los apartamentos y su información.

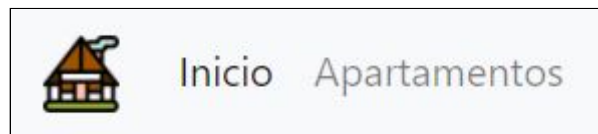


Figura 93: Barra de navegación para usuarios no identificados

A.1.1 Consultar Oferta

Para consultar la oferta actual de la página web, lo único que tenemos que hacer es ir al apartado Inicio de la web y presionar el botón “Oferta Disponible” que se encuentra en el panel de oferta y búsqueda de reservas (Figura 94).

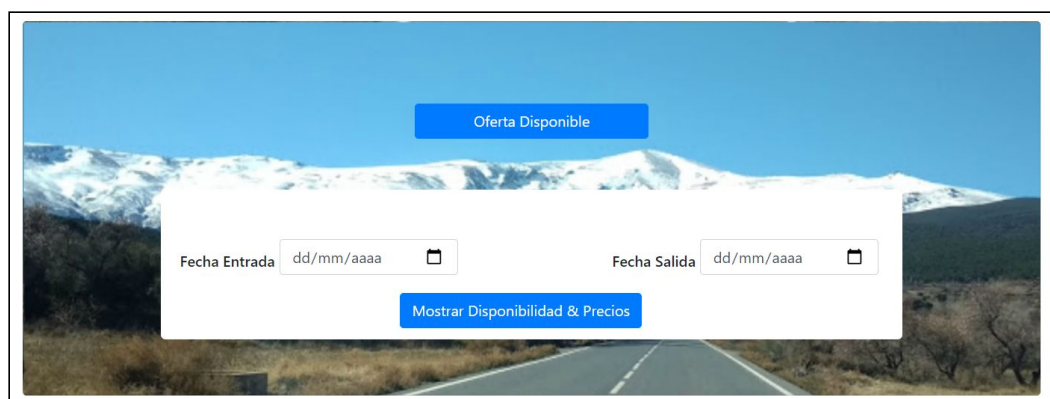


Figura 94: Panel de Oferta y Búsqueda de Reservas

El sistema nos mostrará un Pop Up con la información de la oferta actual (Figura 95).

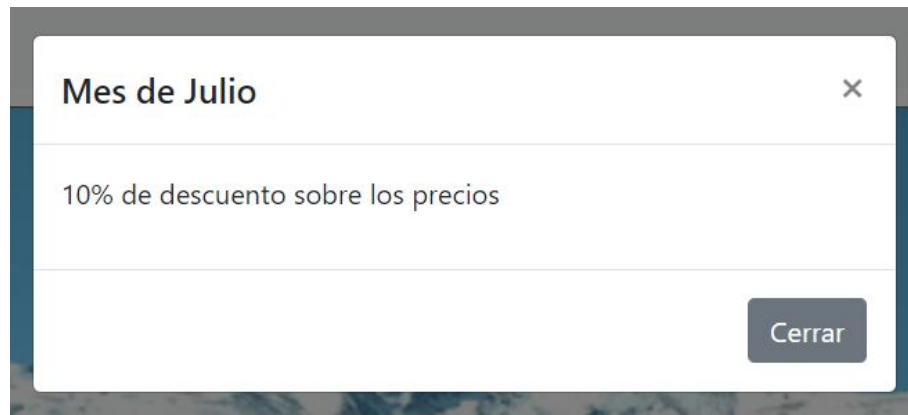


Figura 95: Pop Up de la oferta actual de la web

A.1.2 Reservar Apartamentos

Para realizar una reserva de apartamentos, lo primero que haremos será situarnos en la pestaña de Inicio y, en el panel de oferta y búsqueda de apartamentos (Figura 95). Una vez ahí, introducimos dos fechas en las cuales queremos alquilar apartamentos y presionamos el botón 'Mostrar Disponibilidad & Precios' (Figura 96).

A screenshot of a reservation form. It has two input fields: 'Fecha Entrada' with the value '16/10/2020' and a calendar icon, and 'Fecha Salida' with the value '20/10/2020' and a calendar icon. Below these fields is a blue button with the text 'Mostrar Disponibilidad & Precios'.

Figura 96: Formulario de fechas para reserva de apartamentos

El sistema nos ofrecerá los apartamentos que podemos alquilar, con las plazas que incluye cada uno y el precio total de la estancia (Figura 97). En este momento, si las plazas no son suficientes con un apartamento, debemos seleccionar tantos apartamentos como necesitamos.

A screenshot of a selection interface titled 'Seleccione los apartamentos que desea reservar:'. It lists two options: 'Picón - 2 Plazas (50€)' with a checked checkbox, and 'Postero - 6 Plazas (50€)' with an unchecked checkbox. Each option includes a block of Lorem Ipsum placeholder text. To the right of the text are two photographs of apartment interiors. Below the first option, there is a label 'Dormitorio 1' with two person icons. The second option also has a photograph of an interior.

Figura 97: Posibilidades de apartamentos a elegir en el proceso de reserva

Una vez seleccionados los apartamentos que queremos alquilar, presionamos el botón ‘Reservar’ que se encuentra al final de la página. El sistema ahora nos mostrará los detalles de nuestra reserva y el precio total.

Ya que no estamos reservando como un usuario conectado, nos pedirá nuestros datos y un correo electrónico y contraseña con el cual se creará la cuenta de usuario en la aplicación web (Figura 98).

Detalles de la Reserva

Podrá ver el estado de su reserva en 'Mis Reservas', accediendo con su correo electrónico y su contraseña.

DNI:	Correo Electrónico:	Fecha de Entrada:
<input type="text"/>	<input type="text"/>	<input type="text" value="16/10/2020"/>
Nombre:	Fecha de Nacimiento:	Fecha de Salida:
<input type="text"/>	<input type="text" value="dd/mm/aaaa"/>	<input type="text" value="20/10/2020"/>
Apellidos:	Niños:	Apartamentos: Picón
<input type="text"/>	<input type="text"/>	
Teléfono:	Adultos:	
<input type="text"/>	<input type="text"/>	

Hemos detectado que no estás registrado. Introduce una contraseña para poder acceder a tus reservas y ver el estado en el que se encuentra.

Contraseña:

Precio Total: 50€

[Confirmar Reserva](#)

Figura 98: Formulario de reserva de apartamentos para usuarios no identificados

Completamos los datos correctamente y pulsamos en el botón de ‘Confirmar Reserva’. Si todos los datos están correctos, el sistema nos redirigirá a la página principal. La reserva se habrá realizado y nuestro usuario ha sido creado, por lo que ya podemos conectarnos como un usuario registrado.

A.1.3 Identificarse como Usuario

Una vez hemos realizado nuestra primera reserva y creado nuestro usuario, podemos conectarnos ingresando el correo electrónico y la contraseña que utilizamos en el proceso de reserva. Para ello, presionamos el botón ‘Conéctate’ situado en la barra de navegación.

Cuando pulsamos, se abre un desplegable donde introduciremos los datos y pulsaremos en ‘Entrar’ (Figura 99).



Conéctate

Correo Electrónico:

prueba@prueba.com

Contraseña:

.....

Entrar

Figura 99: Formulario de identificación de usuario

Si todo ha ido bien, estamos conectados como usuario y tenemos acceso a nuevas páginas dentro de la aplicación, dependiendo del rol que tengamos como usuario.

A.1.4 Visualizar Apartamentos

Para visualizar los apartamentos de la web tendremos que ir a la página “Apartados” de la barra de navegación (Figura 93).

Postero

Lorem Ipsum es simplemente el texto de relleno de las imprentas y archivos de texto. Lorem Ipsum ha sido el texto de relleno estándar de las industrias desde el año 1500, cuando un impresor (N. del T. persona que se dedica a la imprenta) desconocido usó una galería de textos y los mezcló de tal manera que logró hacer un libro de textos especimen. No sólo sobrevivió 500 años, sino que tambien ingresó como texto de relleno en documentos electrónicos, quedando esencialmente igual al original. Fue popularizado en los 60s con la creación de las hojas “Letraset”, las cuales contenian pasajes de Lorem Ipsum, y más recientemente con software de autoedición, como por ejemplo Aldus PageMaker, el cual incluye versiones de Lorem Ipsum.

[Ir a Postero](#)



Tizná

Lorem Ipsum es simplemente el texto de relleno de las imprentas y archivos de texto. Lorem Ipsum ha sido el texto de relleno estándar de las industrias desde el año 1500, cuando un impresor (N. del T. persona que se dedica a la imprenta) desconocido usó una galería de textos y los mezcló de tal manera que logró hacer un libro de textos especimen. No sólo sobrevivió 500 años, sino que tambien ingresó como texto de relleno en documentos electrónicos, quedando esencialmente igual al original. Fue popularizado en los 60s con la creación de las hojas “Letraset”, las cuales contenian pasajes de Lorem Ipsum, y más recientemente con software de autoedición, como por ejemplo Aldus PageMaker, el cual incluye versiones de Lorem Ipsum.

[Ir a Tizná](#)



Figura 100: Vista general de la sección de Apartamentos de la web

La aplicación nos redirigirá a la página de los apartamentos, donde podremos ver la imagen principal de estos, una descripción larga y el nombre. Para ver más detalles, presionamos el botón del apartamento al que queremos acceder “Ir a Nombre del Apartamento”.



Figura 101: Vista individual de un apartamento

En esta vista podremos ver una información más detallada del apartamento como las habitaciones, las camas que incluye y un slider con todas las imágenes del mismo (Figura 101).

Además, comentar que hay otra forma de acceder a las páginas de los apartamentos desde una vista en miniatura en forma de tarjetas, situadas en el apartado “Inicio” de la barra de navegación (Figura 102).

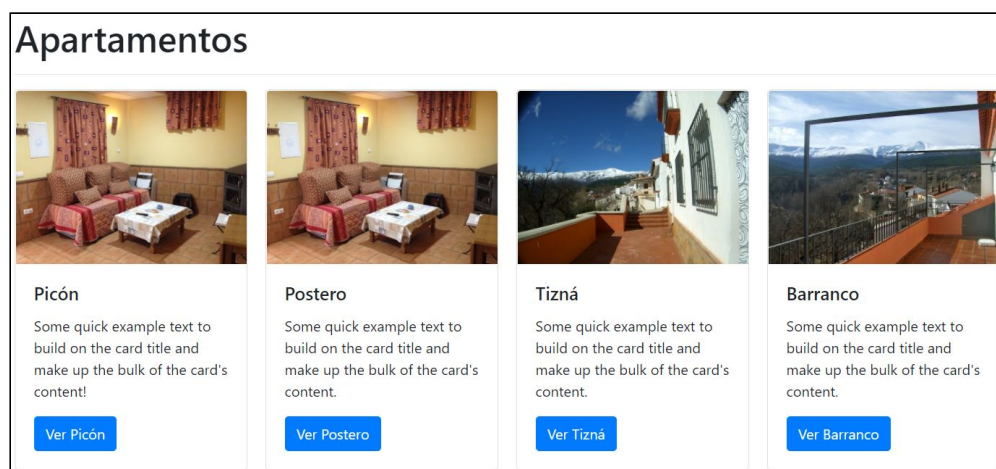


Figura 102: Tarjetas de los apartamentos en la sección de Inicio de la web

A.1.5 Visualizar Calendario de Disponibilidad

Antes de realizar una reserva, podemos visualizar el calendario de disponibilidad desde la sección “Inicio” situada en la barra de navegación (Figura 93).

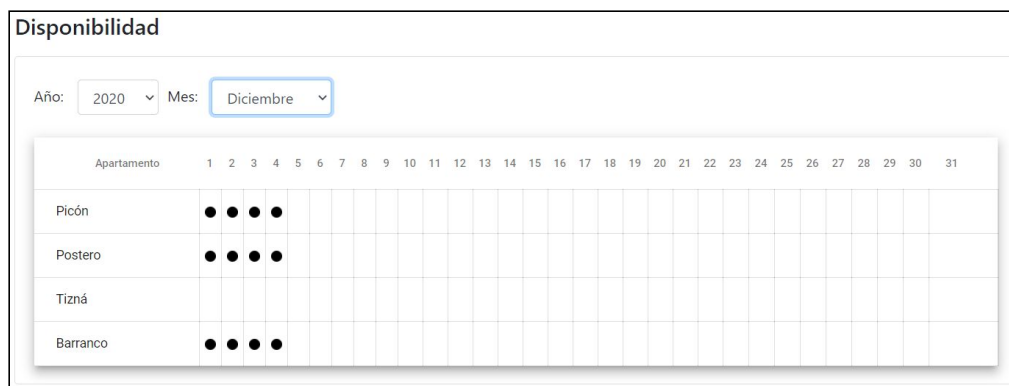


Figura 103: Calendario de Disponibilidad

Encontraremos una tabla compuesta por filas y columnas (Figura 103). Las filas son los apartamentos y las columnas son los días del mes. Para ver la disponibilidad de la web, basta con seleccionar el año y el mes deseados y la tabla cargará los días. Si el día está ocupado con un punto negro, significará que ese día dicho apartamento se encuentra ocupado.

A.2 Usuario Identificado

Cuando entramos a la web como usuario identificado tenemos acceso a un apartado más de la web que hasta ahora estaba oculto en la barra de navegación, la página de “Mis Reservas”. En ella podremos ver información de nuestras reservas, así como completar cuestionarios de satisfacción.

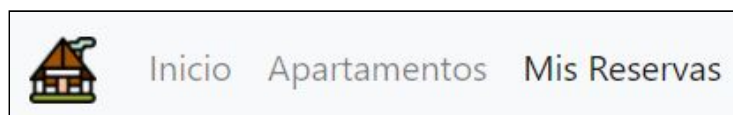


Figura 104: Barra de navegación para usuarios identificados

A.2.1 Visualizar Reservas Usuario

Para ver información sobre nuestras reservas, presionaremos el apartado “Mis Reservas” situado en la barra de navegación (Figura 104).

The screenshot displays a user's reservation list. It contains two reservation cards. Each card shows the apartment name 'Apartamentos: Picón, Postero', the dates, the price (270€), the advance payment (20€), and the status 'Estado: Confirmada' with a checkmark icon. The first reservation has a button labeled 'Completar Cuestionario de Satisfacción'. The second reservation has a green message box at the bottom that says '¡Muchas gracias por completar el cuestionario de satisfacción, nos ayuda a mejorar!'.

Reserva	Apartamentos	Fecha	Precio	Anticipo	Estado	Acción
1	Picón, Postero	Desde 03/04/2020 hasta 04/04/2020	270€	20€	Confirmada ✓	Completar Cuestionario de Satisfacción
2	Picón, Postero	Desde 23/05/2020 hasta 28/05/2020	270€		Confirmada ✓	¡Muchas gracias por completar el cuestionario de satisfacción, nos ayuda a mejorar!

Figura 105: Vista de las reservas de un usuario identificado

El sistema nos proporcionará la información acerca de las reservas que tenemos; la fecha en la que comienzan y finalizan, los apartamentos que incluye, el precio total de la reserva, la cantidad a anticipar y el estado de la misma. Además, si la reserva ha pasado de fecha y su estado es ‘confirmada’, tendremos la opción de completar un cuestionario de satisfacción (Figura 105).

A.2.2 Completar Cuestionario de Satisfacción

Cuando la fecha de una reserva ha pasado y el estado de la reserva quedó como ‘confirmada’, lo que quiere decir que la reserva se realizó correctamente, el sistema nos da la opción de rellenar un cuestionario de satisfacción desde la página ‘Mis Reservas’.

Primeramente presionamos el botón ‘Completar Cuestionario de Satisfacción’ situado al lado de la reserva en concreto (Figura 105). El sistema nos enviará a una página donde introduciremos una puntuación para limpieza, otra para la calidad y un campo de texto como comentario adicional (Figura 106). Completamos los datos y pulsamos el botón de ‘Enviar’, que nos llevará a la página de nuestras reservas.

Formulario para el cuestionario de satisfacción. Incluye tres campos de entrada: 'Limpieza' (con un menú desplegable), 'Calidad' (con un menú desplegable), y 'Comentario' (un área de texto grande). Debajo de los campos hay un botón azul que dice 'Enviar'.

Figura 106: Formulario para el cuestionario de satisfacción

A.2.3 Realizar Reserva Usuarios Identificados

El proceso para reservar apartamentos para un usuario identificado es prácticamente igual que para un usuario no registrado. Primeramente introducimos las fechas de inicio y salida y buscamos las posibilidades, exactamente igual que si no estuviéramos identificados.

Pantalla de búsqueda de posibilidades de reserva con anticipo. En la parte superior hay un mensaje amarillo que indica: "Las fechas introducidas para reserva están comprendidas en un periodo de bastante demanda, por lo que debe tener en cuenta que para que la reserva se confirme deberá realizar una transferencia del 20% de la cuantía total." Debajo, el título "Seleccione los apartamentos que desea reservar:" precede a una lista de opciones. La primera opción es "Picón - 2 Plazas (50€)" con un icono de coche. A la izquierda de esta opción hay un bloque de texto de Lorem Ipsum. A la derecha hay una fotografía de un dormitorio con una cama, una alfombra y una estufa. Debajo de la opción, hay un botón que dice "Dormitorio 1" con un icono de coche.

Figura 107: Búsqueda de posibilidades de reserva con anticipo

De la misma forma, se nos mostrarán las posibilidades a escoger, seleccionaremos uno o más apartamento y presionaremos en el botón reservar.

Detalles de la Reserva

Tenga en cuenta que al ser una reserva anticipada, debe abonar la cantidad del 20% del coste total al número de cuenta que le facilitaremos a continuación. Podrá ver el estado de su reserva en 'Mis Reservas', accediendo con su correo electrónico y su contraseña.

DNI:	Correo Electrónico:	Fecha de Entrada:
44651189R	usuario@usuario.com	16/09/2020
Nombre:	Fecha de Nacimiento:	Fecha de Salida:
Usuario	31/08/1997	20/09/2020
Apellidos:	Niños:	Apartamentos: Picón
1		
Teléfono:	Adultos:	
666666666		

Anticipo a abonar en cuenta: 10€
 Cantidad a abonar de forma presencial: 40€
 Precio Total: 50€

Confirmar Reserva

Figura 108: Formulario de confirmación de reserva para usuarios identificados con anticipo

El sistema rellenará algunos campos automáticamente, como el de DNI, Usuario, Apellidos, Fecha de nacimiento, Email y Teléfono. Los campos estarán desactivados para modificación, exceptuando el del correo electrónico y el teléfono, en caso de que se quiera cambiar la información de contacto (Figura 108).

Además, si la reserva requiere anticipo, el sistema mostrará, además del precio total, la cantidad a abonar en cuenta bancaria y la diferencia que se entregará en mano. Por último, confirmamos la reserva y, si todo está correcto, nos enviará a la página principal y la reserva se habrá realizado. Podremos ver la información en “Mis Reservas”.

A.3 Usuario Administrador

Cuando entramos a la web como usuario administrador tenemos acceso a un último apartado más de la web que hasta ahora estaba oculto en la barra de navegación, el “Panel de Administrador” (Figura 109).



Figura 109: Barra de navegación para usuarios administradores

Desde el Panel de Administrador (Figura 110) podremos gestionar todo lo necesario para administrar la aplicación web; gestión de reservas, oferta, fechas con anticipo, apartamentos, tarifas, periodos y disponibilidad.



Figura 110: Menú de navegación del Panel de Administrador

A.3.1 Gestión de Reservas

Dentro de la gestión de reservas podemos visualizar la información de las reservas existentes y la ficha del cliente a la que pertenece dicha reserva (Figura 111).

Reservas Cuestionarios Oferta & Anticipos Apartamentos Tarifas Disponibilidad

DNI o Pasaporte

Apartamentos: Picón, Postero

Fecha: Desde el 03/04/2020 hasta el 04/04/2020

Adultos: 1

Menores: 1

Estado:

Valoración:

Cliente

Nombre: Usuario

DNI: 44651189R

Teléfono: 666666666

E-mail: usuario@usuario.com

F. Nacimiento: 31/08/1997

Figura 111: Página de gestión de reservas para administradores

Podemos encontrar un filtro que nos permite buscar reservas por DNI de cliente, lo cual mostrará solamente aquellas reservas que haya hecho un cliente con un DNI en particular. Para ello, introducimos el DNI en cuestión y pulsamos en “Buscar” (Figura 112).

66666666R Buscar

Apartamentos: Postero
Fecha: Desde el 20/05/2020 hasta el 23/05/2020
Adultos: 2
Menores: 3
Estado: Confirmada
Valoración: 5

Cliente
Nombre: Joselito
DNI: 66666666R
Teléfono: 666666666
E-mail: joselito@gmail.com
F. Nacimiento: 09/01/1997

Guardar Cambios

Apartamentos: Tizná
Fecha: Desde el 20/05/2020 hasta el 23/05/2020
Adultos: 4
Menores: 1
Estado: Confirmada

Cliente
Nombre: Joselito
DNI: 66666666R
Teléfono: 666666666
E-mail: joselito@gmail.com

Figura 112: Lista de reservas filtradas por DNI

Además, podemos modificar el estado de una reserva o la valoración. La valoración sólo podrá verse desde el panel de administrador, y está diseñada para que el propio administrador tenga una anotación del estado del apartamento tras la reserva.

Introducimos los cambios y pulsamos en “Guardar Cambios”, lo que actualizará la reserva en la base de datos y nos mostrará un Pop Up para confirmar que todo ha ido correctamente (Figura 113).

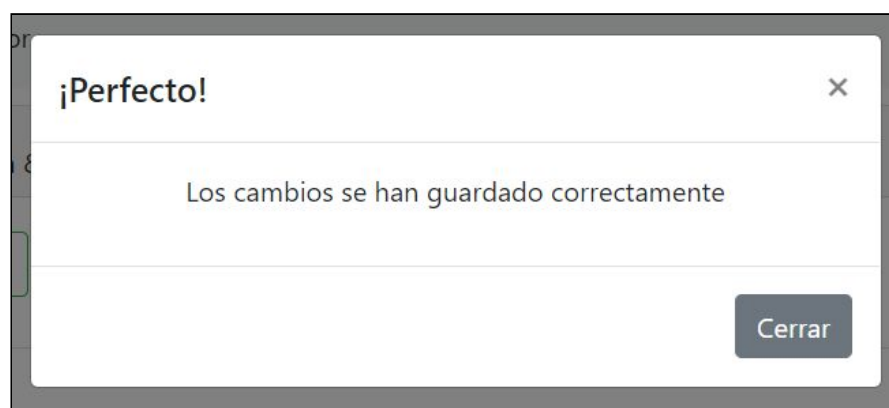


Figura 113: Pop Up de confirmación de cambios

A.3.2 Gestión de Cuestionarios

Dentro de la gestión de cuestionarios podemos visualizar la información de los cuestionarios de satisfacción rellenados por los clientes (Figura 114).

The screenshot shows a web application interface with a top navigation bar containing tabs: Reservas, Cuestionarios (active), Oferta & Anticipos, Apartamentos, Tarifas, and Disponibilidad. Below the navigation bar, there is a search section with a dropdown menu showing 'Picón' and a green 'Buscar' button. The main content area displays two survey entries. The first entry is for 'Apartamentos: Picón, Postero', with a date range 'Desde el 23/05/2020 hasta el 28/05/2020' and 'Cliente: Usuario'. It shows 'Calidad: 3', 'Limpieza: 8', and a 'Comentario' box containing the text 'Todo muy rústico.' The second entry is for 'Apartamentos: Postero', with a date range 'Desde el 05/07/2020 hasta el 10/07/2020' and 'Cliente: Admin'. It shows 'Calidad: 5', 'Limpieza: 7', and a 'Comentario' box containing the text 'NULL'.

Figura 114: Página de gestión de cuestionarios de satisfacción

Además, podemos filtrar los cuestionarios por apartamento, mostrando sólo aquellos cuestionarios de aquellas reservas que estén asociadas a un apartamento concreto. Para ello, en el desplegable (Figura 114), seleccionamos el apartamento en cuestión y presionamos el botón “Buscar”. El sistema filtrará los resultados y los mostrará por pantalla (Figura 115).

This screenshot is identical to the one in Figure 114, showing the same survey entries. However, the 'Buscar' button in the search section is highlighted with a green border, indicating it has been clicked or is the focus of the action.

Figura 115: Filtrado de cuestionarios de satisfacción por apartamento

A.3.3 Gestión de Oferta y Anticipos

Dentro de la gestión de oferta y anticipos podemos modificar la oferta actual o desactivarla, así como añadir o quitar intervalos de fechas que requieren un anticipo.



Oferta

Título:

Descripción:

Activada: ☒

Actualizar Oferta

Figura 115: Formulario de actualización de la oferta

Para modificar la oferta actual, basta con modificar los campos deseados y presionar el botón “Actualizar Oferta” (Figura 116). Si desactivamos la oferta, esta no estará visible en la página de “Inicio”.

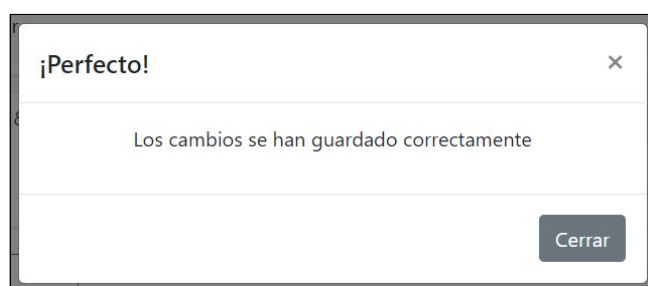
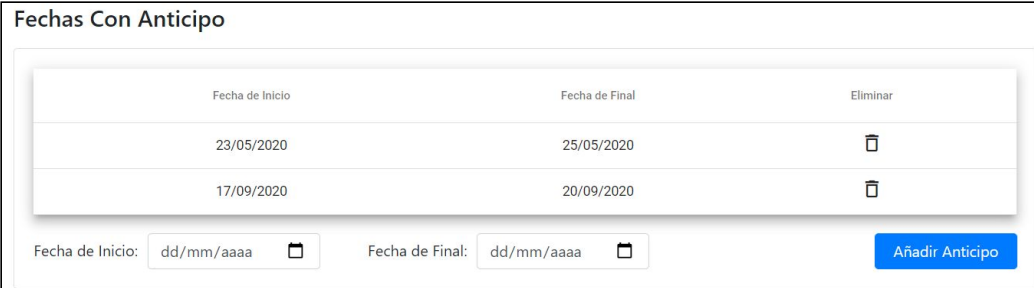


Figura 116: Pop Up de confirmación de cambios guardados

Si todo ha ido correctamente, el sistema mostrará un Pop Up de confirmación (Figura 116).



Fechas Con Anticipo

Fecha de Inicio	Fecha de Final	Eliminar
23/05/2020	25/05/2020	
17/09/2020	20/09/2020	

Fecha de Inicio:

Fecha de Final:

Añadir Anticipo

Figura 117: Panel de gestión de intervalos de Anticipo

En la misma página podemos tanto crear como eliminar intervalos de fechas que requieren un anticipo.

Para eliminar un intervalo, basta con presionar el icono de papelera que se encuentra en la tabla, en la fila de ese periodo concreto (Figura 117). El sistema mostrará un Pop Up para confirmar que se ha eliminado correctamente (Figura 116).

Para agregar un nuevo intervalo, introducimos la fecha de inicio y fin en el formulario que se encuentra debajo de la tabla y presionamos el botón de “Añadir Anticipo” (Figura 118).



Formulario de creación de anticipo. Incluye campos para 'Fecha de Inicio' (17/09/2020) y 'Fecha de Final' (30/09/2020), cada uno con un icono de calendario. A la derecha hay un botón azul que dice 'Añadir Anticipo'.

Figura 118: Formulario de creación de creación de anticipo

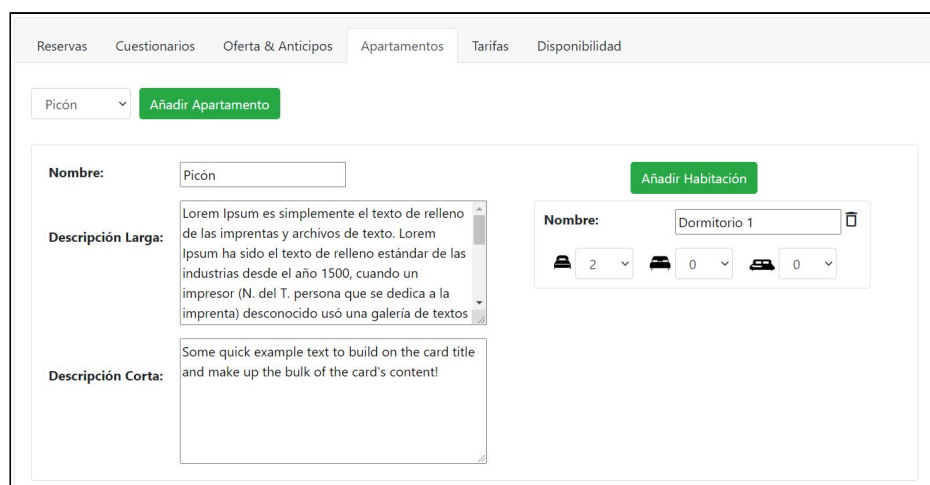
Si todo ha ido bien, se mostrará un Pop Up de confirmación y el nuevo anticipo será añadido en la tabla (Figura 119). El sistema lo tendrá en cuenta cuando un cliente realice una reserva dentro de ese intervalo.

Fecha de Inicio	Fecha de Final	Eliminar
23/05/2020	25/05/2020	
17/09/2020	30/09/2020	

Figura 119: Tabla de intervalos de anticipo

A.3.4 Gestión de Apartamentos

Dentro de la página de gestión de apartamentos podemos crear, eliminar y modificar apartamentos en la base de datos.



Panel de gestión de apartamentos. En la parte superior hay una barra de navegación con pestañas: Reservas, Cuestionarios, Oferta & Anticipos, Apartamentos (seleccionada), Tarifas y Disponibilidad. Debajo de las pestañas hay un menú desplegable con 'Picón' y un botón verde 'Añadir Apartamento'. El formulario principal tiene tres secciones: 'Nombre' con un campo de texto 'Picón'; 'Descripción Larga' con un área de texto que contiene Lorem Ipsum; y 'Descripción Corta' con un área de texto que contiene 'Some quick example text to build on the card title and make up the bulk of the card's content!'. A la derecha de estas secciones hay un sub-formulario para 'Añadir Habitación' con un botón verde. Este sub-formulario incluye un campo 'Nombre' con 'Dormitorio 1' y tres iconos de camas con valores numéricos: 2, 0 y 0.

Figura 120: Panel de gestión de apartamentos

En el menú desplegable podemos seleccionar un apartamento a editar (Figura 120). Una vez dentro, podemos modificar sus campos, añadir y eliminar imágenes, cambiar la imagen principal, quitar habitaciones presionando el botón de icono de la papelera frente a la habitación y añadir una nueva habitación pulsando en el botón “Añadir Habitación”.

Para subir una imagen, bajamos al apartado de galería y pulsamos “Seleccionar archivo” para buscar una imagen de nuestro ordenador. Una vez seleccionada la imagen, pulsamos en el botón “Subir” y la imagen estará en la galería (Figura 121).

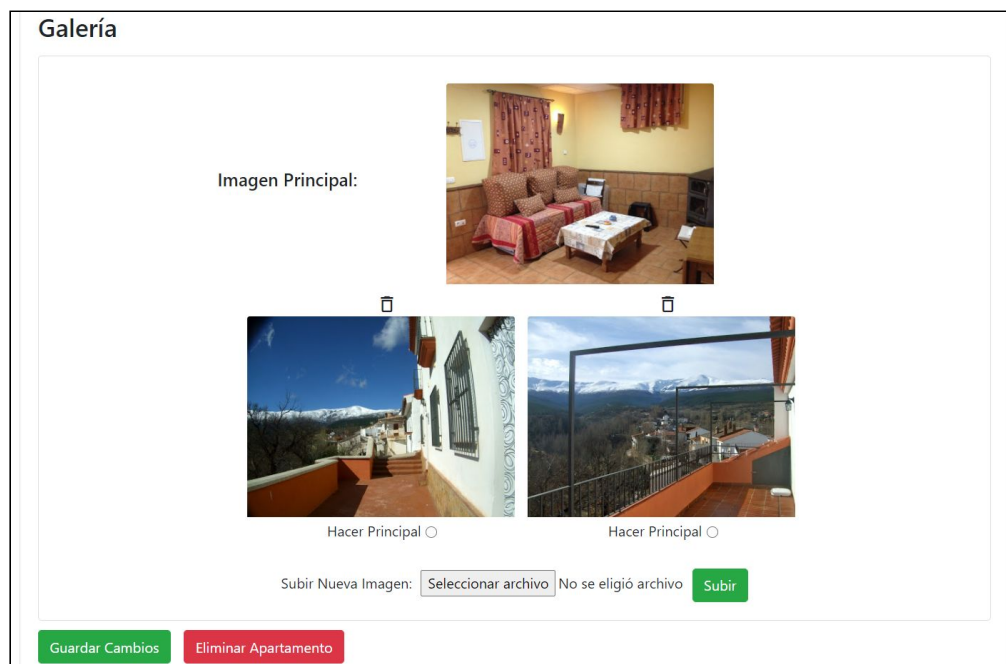


Figura 121: Panel para la gestión de la galería de un apartamento

El sistema también nos permite eliminar fotos presionando el icono de la papelera situado encima de la imagen, o cambiar la imagen principal que se verá en la página de visualización de apartamentos presionando el botón “Hacer Principal” de la imagen deseada.

Una vez modificado los campos deseados del apartamento, pulsamos en el botón “Guardar Cambios” (Figura 121) y, si todo ha ido bien, el sistema nos mostrará un Pop Up de que los cambios han sido guardados correctamente (Figura 116).

Para eliminar un apartamento, basta con situarse en la página de edición del apartamento a eliminar y presionar el botón “Eliminar Apartamento” situado al final de la página. El sistema nos preguntará si estamos seguros de que queremos eliminar el apartamento (Figura 122). Si confirmamos, nos mostrará un Pop Up de confirmación de cambios y nos cargará la página de edición del siguiente apartamento.

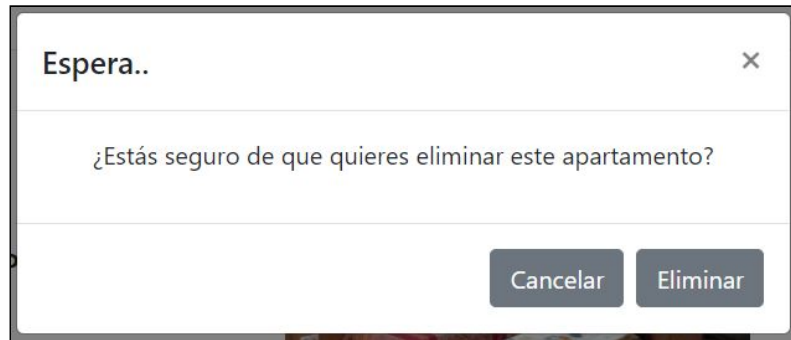


Figura 122: Pop Up de confirmación de eliminación de apartamento

Para crear un apartamento, presionamos el botón de “Añadir Apartamento” situado en la parte superior de la página (Figura 120), lo que nos llevará a la página de creación de apartamento (Figura 123).

A screenshot of a web form for creating an apartment. At the top, there is a navigation bar with tabs: "Reservas", "Cuestionarios", "Oferta & Anticipos", "Apartamentos" (which is active), "Tarifas", and "Disponibilidad". The form has three input fields: "Nombre:" (a single-line text box), "Descripción Larga:" (a multi-line text area), and "Descripción Corta:" (a multi-line text area). At the bottom left of the form is a green button labeled "Crear Apartamento".

Figura 123: Formulario de creación de apartamento

Una vez introducidos los datos, presionamos el botón “Crear Apartamento” y el sistema nos mostrará un Pop Up para notificarnos que todo ha ido correctamente. Nos redirigirá a la página de edición de apartamentos para seguir con su configuración.

A.3.5 Gestión de Tarifas y Periodos

En el panel de gestión de tarifas y periodos podemos visualizar, crear y eliminar tarifas para cada apartamento, de la misma forma que podemos visualizar, crear y eliminar periodos para cada tarifa. Lo primero que haremos será seleccionar el apartamento para el cual queremos visualizar las tarifas en la parte superior de la página (Figura 124).

Noches	Precio Diario	Precio Finde	Eliminar Tarifa
2	8	10	
3	10	12	
4	12	14	
6	3	4	

Noches: Precio Diario: Precio Finde: Añadir Tarifa

Figura 124: Tabla de tarifas y formulario de creación de tarifas

Para eliminar una tarifa, basta con pulsar el icono de papelera situado en la fila de la tarifa que queremos eliminar. El sistema mostrará un Pop Up indicando que todo ha ido correctamente (Figura 113).

Para crear una tarifa, rellenaremos el formulario que se encuentra debajo de la tabla de tarifas indicando las noches de la tarifa, el precio en día de diario y el precio para el fin de semana. Seguidamente, pulsaremos el botón de “Añadir Tarifa” y la tarifa será añadida correctamente a la tabla. Podremos ver un Pop Up que nos notificará de que los cambios se han realizado correctamente (Figura 113).

Fecha Comienzo	Precio Final	Suma	Eliminar Periodo
21/12/2019	09/01/2020	5	
21/12/2019	09/01/2020	-3	
01/07/2020	31/07/2020	-10	
23/06/2020	27/06/2020	-5	

Tarifa: 2 Noches

Fecha de Comienzo: Fecha de Final: Suma: Añadir Periodo

Figura 125: Tabla de periodos por tarifa y formulario de creación de periodos

Para la gestión de los periodos el sistema es muy similar. Primeramente seleccionaremos la tarifa para la cual queremos visualizar los periodos y la tabla se actualizará.

Para eliminar un periodo, basta con pulsar el icono de papelera situado en la fila del periodo que queremos eliminar. El sistema mostrará un Pop Up indicando que todo ha ido correctamente (Figura 113).

Para crear un periodo, rellenaremos el formulario que se encuentra debajo de la tabla de periodos indicando la fecha de comienzo, la fecha de fin y la suma. Seguidamente, pulsaremos el botón de “Añadir Periodo” y el periodo será añadido correctamente a la tabla. Podremos ver un Pop Up que nos notificará de que los cambios se han realizado correctamente (Figura 113).

A.3.6 Gestión de Disponibilidad

En la pestaña de disponibilidad, al igual que un usuario normal, podremos ver una tabla en la que seleccionaremos un mes y un año y el sistema mostrará qué días están ocupados los apartamentos.

La diferencia con respecto a un usuario normal es que si somos administradores, al presionar uno de los días, podremos indicar que ese día manualmente sin necesidad de que exista una reserva, o bien eliminar esos días para que el sistema entienda que están libres para su reserva.

Reservas

Cuestionarios

Oferta & Anticipos

Apartamentos

Tarifas

Disponibilidad

Disponibilidad

Año: 2020

Mes: Diciembre

Apartamento	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Picón	●	●	●	●																											
Postero	●	●	●	●																											
Tizná																															
Barranco	●	●	●	●																											

Figura 125: Tabla de disponibilidad del panel de administrador

Pulsando un día ocupado, el sistema eliminará ese día como un día reservado y actualizará la tabla, mostrando un Pop Up para notificarnos de que todo ha ido correctamente. Cuando cerremos el Pop Up, comprobaremos que ese día ya cargará un punto negro.

De la misma forma, si pulsamos un día libre, el sistema añadirá ese día como día reservado, mostrando un Pop Up para notificarnos que todo ha ido correctamente y añadiendo el punto negro en ese día para ese apartamento.



UNIVERSIDAD
DE MÁLAGA

| **uma.es**